

TUGAS AKHIR - KS141501

# ANALISIS SENTIMEN TERHADAP KINERJA PELAYANAN PUBLIK DI KOTA SURABAYA BERDASARKAN KLASIFIKASI KOMENTAR DI MEDIA SOSIAL DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES

NUKE YULNIDA ADEN FARADHILLAH  
NRP 5212 100 123

Dosen Pembimbing  
Renny Pradina Kusumawardani, S.T, M.T  
Irmasari Hafidz, S.Kom, M.Sc

JURUSAN SISTEM INFORMASI  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



FINAL PROJECT - KS141501

# ANALYSIS OF SENTIMENTALITY TOWARDS THE PERFORMANCE OF PUBLIC SERVICES IN SURABAYA BASED ON COMMENTARY CLASSIFICATION IN SOCIAL MEDIA USING NAÏVE BAYES ALGORITHM

NUKE YULNIDA ADEN FARADHILLAH  
NRP 5212 100 123

Supervisor

Renny Pradina Kusumawardani, S.T, M.T  
Irmasari Hafidz, S.Kom, M.Sc

DEPARTMENT OF INFORMATION SYSTEM  
Faculty of Information Technology  
Institute of Technology Sepuluh Nopember  
Surabaya 2016

## LEMBAR PENGESAHAN

### **ANALISIS SENTIMEN TERHADAP KINERJA PELAYANAN PUBLIK DI KOTA SURABAYA BERDASARKAN KLASIFIKASI KOMENTAR DI MEDIA SOSIAL DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES**

### **TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**NUKE YULNIDA ADEN FARADHILLAH**  
NRP. 5212 100 123

Surabaya, 20 Juni 2016

**KETUA  
JURUSAN SISTEM INFORMASI**

**Dr. Ir. Aris Tjahyanto, M. Kom.**  
NIP. 19650310 199102 001

## LEMBAR PERSETUJUAN

### **ANALISIS SENTIMEN TERHADAP KINERJA PELAYANAN PUBLIK DI KOTA SURABAYA BERDASARKAN KLASIFIKASI KOMENTAR DI MEDIA SOSIAL DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES**

#### **TUGAS AKHIR**


Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**NUKE YULNIDA ADEN FARADHILLAH**  
**NRP. 5212 100 123**

Disetujui Tim Penguji : Tanggal Ujian : 9 Juni 2016  
Periode Wisuda : September 2016

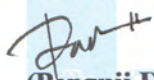
**Renny Pradina K., S.T, M.T**

  
(Pembimbing I)

**Irmasari Hafidz, S.Kom, M.Sc**

  
(Pembimbing II)

**Radityo Prasetyanto W., S.Kom, M.Kom**

  
(Penguji I)

**Nur Aini Rakhmawati, S.Kom., M.Sc**

  
(Penguji II)

# **ANALISIS SENTIMEN TERHADAP KINERJA PELAYANAN PUBLIK DI KOTA SURABAYA BERDASARKAN KLASIFIKASI KOMENTAR DI MEDIA SOSIAL DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES**

**Nama Mahasiswa** : NUKE YULNIDA ADEN F.  
**NRP** : 5212 100 123  
**Jurusan** : SISTEM INFORMASI FTIF-ITS  
**Dosen Pembimbing I** : Renny Pradina K. S.T, M.T  
**Dosen Pembimbing II** : Irmasari Hafidz, S.Kom, Msc

## **ABSTRAK**

*Salah satu sumber informasi yang dibutuhkan oleh Pemerintah untuk dapat meningkatkan kinerjanya adalah umpan balik dari masyarakat. Di era sekarang, media sosial seringkali digunakan oleh masyarakat sebagai alat untuk menyampaikan keluhan terhadap pelayanan yang telah diberikan, memberikan usulan terhadap program kerja atau sekedar ingin mengetahui informasi terkini. Hasil opini yang telah diberikan oleh masyarakat inilah yang dapat digunakan sebagai bahan evaluasi oleh pemilik perusahaan atau instansi terkait untuk melakukan perbaikan dan juga dapat meningkatkan kualitas kinerja yang baik ke depannya. Di Kota Surabaya, terdapat berbagai media sosial yang dapat dipergunakan masyarakat untuk menyampaikan semua keluhan, informasi, maupun saran terhadap proses pembangunan kota. Salah satu media sosial yang sering digunakan masyarakat yaitu twitter. Untuk dapat memantau seluruh tweet yang ada pada*



*seluruh akun twitter bukanlah sesuatu yang mudah untuk dilakukan secara manual.*

*Untuk itu dalam penelitian ini akan dilakukan pengklasifikasian opini masyarakat yang terkandung dalam media twitter secara otomatis. Pengklasifikasian dilakukan dengan menggunakan algoritma Naïve Bayes dan Support Vector Machine (SVM). Adapun data yang digunakan yaitu mengambil dari twitter @e100ss dan @SapawargaSby. Data dari twitter ini akan diolah dan dikelompokkan menjadi 3 kelas sentimen yaitu positif, negatif, dan netral. Model klasifikasi terbaik didapatkan dengan menggunakan algoritma Support Vector Machine (SVM) dengan hasil akurasi sebesar 79,81%. Untuk nilai precision pada kelas positif, negatif, dan netral berturut-turut yaitu 7,69%, 5,06%, dan 99,8%. Sedangkan nilai recall yaitu 66,66%, 100%, dan 79,67%, nilai f-measure yaitu 13,79%, 9,63%, dan 88,62%. Selain itu, pada penelitian ini juga dilakukan pembuatan visualisasi berupa wordcloud dan grafik streamgraph yang ditampilkan secara interaktif dengan aplikasi berbasis web yaitu Rshiny.*

**Kata kunci : Klasifikasi Teks, Twitter, Algoritma Naïve Bayes, Algoritma Support Vector Machine (SVM), Rshiny**

**ANALYSIS OF SENTIMENTALITY TOWARDS THE  
PERFORMANCE OF PUBLIC SERVICES IN SURABAYA  
BASED ON COMMENTARY CLASSIFICATION IN  
SOCIAL MEDIA USING NAÏVE BAYES ALGORITHM**

**Name : NUKE YULNIDA ADEN FARADHILLAH**  
**NRP : 5212 100 123**  
**Departement : INFORMATION SYSTEM FTIF-ITS**  
**Supervisor 1 : RENNY PRADINA K., S.T, M.T**  
**Supervisor 2 : IRMASARI HAFIDZ S.Kom, M.Sc**

**ABSTRACT**

*In order to improve its performance, one of the most crucial information sources for the Government is the feedback from the society it is serving. Nowadays, people use the social media to voice their aspiration on various subjects, including public service, whether it is complaining about the quality of service, giving suggestions, input, or advice, or merely querying about some latest information. These kinds of tweets comprise the kind of feedback that could be used by the government and the related government bodies to evaluate and improve the quality of its service. For the city of Surabaya, there are many social media outlets that could be used by the people. On of the most popular is Twitter. However, it is infeasible to monitor the entire twitter universe manually for information that is relevant to the level of satisfaction of the people of Surabaya toward the City Government. For this reason, this research explores and implements methods for automatic*

*classification of the opinion of the people of Surabaya that is manifested in tweets. The algorithms explored are Naïve Bayes and Support Vector Machines (SVMs), since these two algorithms are known to give excellent performance in the task of text classification. The data that is used are tweets which mention the accounts @e100ss and @SapawargaSby, since these two accounts are the most popular in Surabaya in terms of being used by the people to express their opinion of the government. The best classification model obtained by using Support Vector Machine (SVM) algorithms with an accuracy result of 79.81%. For precision value in positive, negative, and neutral class is respectively 7.69%, 5.06%, and 99.8%. While the recall value is 66.66%, 100%, and 79.67%, the f-measure value is 13.79%, 9.63% and 88.62%. In this research interactive visualizations of the results have also been produced in the form of wordclouds and streamgraphs. These are then published using a web-based application for data publishing, RShiny.*

**Keyword :** *Text classification, Twitter, Naïve Bayes Algorithm, Support Vector Machine (SVM) Algorithm, Rshiny*



## DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
LEMBAR PERSETUJUAN.....	v
ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR.....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxi
1. BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Tugas Akhir.....	4
1.4. Tujuan Tugas Akhir .....	4
1.5. Manfaat Tugas Akhir .....	5
1.6. Relevansi.....	5
2. BAB II TINJAUAN PUSTAKA .....	7
2.1. Penelitian Sebelumnya.....	7
2.2. Dasar Teori .....	14
2.2.1. Twitter.....	14
2.2.1.1. @SapawargaSby .....	14
2.2.1.2. @e100ss .....	15
2.2.2. Klasifikasi .....	15
2.2.2.1. Klasifikasi Teks.....	16
2.2.3. Model Klasifikasi.....	16
2.2.4. Data Mining .....	17
2.2.5. Text Corpus.....	19
2.2.6. Praproses data .....	20
2.2.7. Algoritma Naive Bayes.....	20
2.2.8. Support Vector Machine(SVM).....	22
2.2.9. Evaluasi performa klasifikasi.....	24
2.2.9.1. Akurasi .....	24
2.2.9.2. Precision, Recall, dan F-Measure.....	25

2.2.9.3.	F-Measure.....	26
3.	BAB III METODE PENELITIAN.....	27
3.1.	Studi Literatur .....	27
3.2.	Pengumpulan Data .....	28
3.3.	Pemrosesan Data .....	30
3.3.1.	Tahap Pelabelan Data .....	30
3.3.2.	Tahap Praproses Data .....	30
3.4.	Uji Coba dan Proses Klasifikasi.....	32
3.4.1.	Tahap pelatihan dengan algoritma Naïve Bayes ..	32
3.4.2.	Evaluasi performa klasifikasi .....	32
3.5.	Pembahasan hasil percobaan.....	32
3.6.	Visualisasi hasil.....	33
3.7.	Pembuatan Buku Laporan .....	33
4.	BAB IV PERANCANGAN .....	35
4.1.	Pengumpulan Data .....	35
4.2.	Penyeleksian Atribut .....	36
4.3.	Pelabelan Kelas Data .....	37
4.4.	Penghapusan Data .....	39
4.5.	Praproses Data.....	39
5.	BAB V IMPLEMENTASI.....	43
5.1.	Data Implementasi .....	43
5.2.	Lingkungan Implementasi.....	44
5.3.	Klasifikasi .....	45
5.3.1.	Skenario Klasifikasi .....	45
5.3.2.	Skenario 1 .....	48
5.3.3.	Skenario 2 .....	57
5.4.	Klasifikasi Data Kinerja Pelayanan Publik Kota Surabaya.....	63
5.5.	Visualisasi .....	64
5.5.1.	Wordcloud.....	65
5.5.2.	Streamgraph .....	65
5.5.3.	Shiny.....	65
6.	BAB VI ANALISA HASIL DAN PEMBAHASAN...71	
6.1.	Analisis Model Klasifikasi.....	71
6.1.1.	Hasil Skenario 1 .....	71

6.1.1.1	Model 1 .....	72
6.1.1.2	Model 2 .....	72
6.1.1.3	Model 3 .....	72
6.1.1.4	Model 4 .....	74
6.1.1.5	Model 5 .....	76
6.1.1.6	Model 6 .....	76
6.1.1.7	Model 7 .....	77
6.1.2.	Kesimpulan algoritma Naive Bayes.....	79
6.1.3.	Hasil Skenario 2 .....	79
6.1.2.1	Model 1 .....	79
6.1.2.2	Model 2 .....	80
6.1.2.3	Model 3 .....	80
6.1.2.4	Model 4 .....	81
6.1.2.5	Model 5 .....	81
6.1.2.6	Model 6 .....	83
6.1.4.	Kesimpulan algoritma SVM .....	83
6.1.5	Data Tentang Surabaya .....	84
6.1.5.2.	Pengerjaan dengan Support Vector Machine.....	85
6.2.	Analisis Performa Model Klasifikasi .....	85
6.2.1	Skenario 1.....	86
6.2.2	Skenario 2.....	87
6.3.	Kinerja Pelayanan Publik di Kota Surabaya .....	88
7.	BAB VII KESIMPULAN DAN SARAN .....	93
7.1.	Kesimpulan .....	93
7.2.	Saran .....	94
8.	DAFTAR PUSTAKA .....	97
9.	RIWAYAT PENULIS .....	101
10.	DAFTAR LAMPIRAN.....	103
A.	LAMPIRAN A.....	A-1
A.1	NAIVE BAYES .....	A-1
A.1.1	Kode program model 1,2,3.....	A-1
A.1.2	Kode program model 4.....	A-2
A.1.3	Kode Program model 5 .....	A-4
A.1.4	Kode program model 6.....	A-7
A.1.5	Kode program model 7a.....	A-10

A.1.6	Kode program model 7b.....	A-13
A.1.7	Kode program model 7c.....	A-16
A.1.8	Kode program model 7d.....	A-19
A.2	SVM.....	A-23
A.2.1	Kode program model 1.....	A-23
A.2.2	Kode program model 2.....	A-25
A.2.3	Kode program model 3.....	A-26
A.2.4	Kode program model 4.....	A-28
A.2.5	Kode program model 5.....	A-30
A.2.6	Kode program model 6.....	A-31
A.3	SHINY.....	A-33
A.3.1.	Ui.r.....	A-33
A.3.2.	Global.R.....	A-35
A.3.3.	Server.R.....	A-42
B.	LAMPIRAN B.....	B-1
B.1	Positif.....	B-1
B.1.1	10.....	B-1
B.1.2	20.....	B-1
B.1.3	30, 40.....	B-1
B.2	Negatif.....	B-2
B.2.1	40.....	B-2
B.2.2	50.....	B-2
B.2.3	60.....	B-3
B.2.4	70, 80, 90.....	B-3
B.2.5	100.....	B-3
B.2.6	110.....	B-3
B.3	Netral.....	B-4
B.3.1	40.....	B-4
B.3.2	60.....	B-5
B.3.3	80.....	B-5
B.3.4	100.....	B-6
B.3.5	120.....	B-6
B.3.6	140.....	B-7
C.	LAMPIRAN C.....	C-1
C.1	Stopword dari github.....	C-1

C.2	Stopword model 6 skenario 1 .....	C-4
C.3	Stopword model 7 ketiga .....	C-6
C.4	20 kata teratas yang akan dijadikan stopwords .....	C-6
D.	LAMPIRAN D.....	D-1
D.1	Kernel Linear 3 Pembobotan .....	D-1
D.2	Kernel Linear Hanya Tf .....	D-4
D.3	Kernel RBF 3 Pembobotan .....	D-6
D.3.1	Binary.....	D-6
D.3.2	Tf-Idf.....	D-8
D.3.3	Tf.....	D-10
D.4	Kernel RBF Hanya Tf.....	D-12
E.	LAMPIRAN E .....	E-1

## DAFTAR TABEL

Tabel 2.1 Ringkasan Studi Sebelumnya.....	8
Tabel 4.1 Atribut data .....	35
Tabel 4.2 Penyeleksian atribut data.....	36
Tabel 4.3 Contoh Data Hasil Seleksi Atribut .....	37
Tabel 4.4 Contoh data yang sudah dilabeli .....	38
Tabel 4.5 Data hasil crawling.....	42
Tabel 5.1 Distribusi Penggunaan Dataset .....	44
Tabel 5.2 Lingkungan implementasi perangkat keras .....	44
Tabel 5.3 Dekscripsi dan tujuan masing-masing model .....	48
Tabel 5.4 Jenis Metode Pembobotan.....	54
Tabel 5.5 Deskripsi dan ytujuan pembuatan skenario 2.....	57
Tabel 5.6 Parameter Optimasi SVM .....	60
Tabel 5.7 Nilai variabel cost dan gamma model 3 .....	62
Tabel 5.8 Nilai variabel cost dan gamma model 4 .....	62
Tabel 5.9 Distribusi Pembagian Data Tentang Surabaya.....	63
Tabel 6.1 Hasil akurasi masing-masing metode pembobotan	72
Tabel 6.2 Hasil Akurasi dengan percobaan frequent word ....	73
Tabel 6.3 Jumlah kata yang muncul.....	73
Tabel 6.4 Hasil Akurasi setelah dibakukan .....	74
Tabel 6.5 Kata yang muncul pada FQ 100.....	75
Tabel 6.6 Hasil Akurasi model 5.....	76
Tabel 6.7 Hasil Percobaan model 6.....	77
Tabel 6.8 Model Pengerjaan model 7.....	78
Tabel 6.9 Hasil Akurasi dengan model 7 .....	78
Tabel 6.10 Hasil Akurasi awal model 1 .....	80
Tabel 6.11 Hasil Akurasi dengan metode yang berbeda .....	80
Tabel 6.12 Perbandingan hasil akurasi dengan Naive Bayes.	84
Tabel 6.13 Perbandingan hasil akurasi dengan SVM.....	85
Tabel 6.14 <i>Confusion matrix</i> terbaik skenario 1 .....	86
Tabel 6.15 Nilai <i>precision</i> , <i>recall</i> , dan <i>f-measure</i> skenario 1	86
Tabel 6.16 <i>Confusion matrix</i> terbaik skenario 2 .....	87
Tabel 6.17 Nilai <i>precision</i> , <i>recall</i> , dan <i>f-measure</i> skenario 2.	87
Tabel 6.18 <i>Confusion matrix</i> data Surabaya .....	89

Tabel 6.19 Nilai <i>precision</i> , <i>recall</i> , dan <i>f-measure</i> data Surabaya .....	89
Tabel 6.20 Isi dari 15 kata teratas masing-masing kelas .....	89
Tabel 6.21 Isi dari 15 kata teratas setelah dilakukan stopwords .....	90
Tabel 6.22 Isi kata dari masing-masing kelas.....	91



## DAFTAR GAMBAR

Gambar 2.1 Model Klasifikasi .....	17
Gambar 2.2 Proses Data Mining .....	18
Gambar 2.3 Support Vector pada Metode SVM .....	23
Gambar 2.4 <i>Confusion matrix</i> .....	25
Gambar 3.1 Alur pengerjaan tugas akhir .....	29
Gambar 3.2 Tahap praproses data .....	31
Gambar 5.1 Alur pengerjaan skenario klasifikasi 1 .....	52
Gambar 5.2 Proses Dihasilkan Data Baru .....	55
Gambar 5.3 Alur pengerjaan skenario klasifikasi 2 .....	60
Gambar 5.4 Tab Wordcloud .....	66
Gambar 5.5 Menu pilihan kelas klasifikasi semua data .....	66
Gambar 5.6 Tab Wordcloud Data Surabaya .....	67
Gambar 5.7 Menu Pilihan Kelas Klasifikasi Data Surabaya ..	68
Gambar 5.8 Tab Streamgraph Semua Data .....	68
Gambar 5.9 Menu Pilihan Kelas Klasifikasi .....	69
Gambar 5.10 Tab Streamgraph Data Surabaya .....	69
Gambar 6.1 Perbandingan hasil antara sudah baku dan belum baku .....	75
Gambar 6.2 Gridsearch dengan Kernel RBF metode Tf .....	82
Gambar 6.3 Gridsearch dengan Kernel RBF metode Binary ..	82
Gambar 6.4 Gridsearch dengan Kernel RBF metode Tf-Idf ..	83

# **BAB I**

## **PENDAHULUAN**

Bab pendahuluan ini membahas tentang latar belakang pengerjaan tugas akhir, rumusan permasalahan yang dihadapi dalam pengerjaan tugas akhir, batasan permasalahan pengerjaan tugas akhir, tujuan pengerjaan tugas akhir, dan manfaat dari pengerjaan tugas akhir.

### **1.1. Latar Belakang Masalah**

Pemerintah merupakan suatu bentuk organisasi yang bekerja dengan tugas menjalankan suatu sistem pemerintahan. Salah satu fungsi tugas dari pemerintah adalah menyebarluaskan informasi, mengomunikasikan kebijakan, rencana kerja, dan capaian kinerja kepada masyarakat luas, melalui media tradisional, media konvensional, dan media baru [1]. Komunikasi dengan menggunakan media baru atau teknologi internet dapat menjangkau langsung dan cepat kepada semua pihak yang saat ini telah dinikmati secara luas. Teknologi internet adalah salah satu bukti yang bisa kita lihat sebagai pesatnya perkembangan teknologi di bidang informasi dan telekomunikasi. Perkembangan internet di Indonesia telah menunjukkan perkembangan yang signifikan. Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) jumlah pengguna internet di Indonesia mencapai 88,1 juta orang hingga akhir tahun 2014 dan mengalami peningkatan sebanyak 6,3% dari tahun 2013 [2].

Dewasa ini kehidupan masyarakat mengalami banyak perubahan akibat dari kemajuan ilmu pengetahuan dan teknologi yang pesat. Masyarakat telah mengalami perubahan pola pikir ke arah yang semakin kritis dalam menanggapi kondisi yang ada, dimana dengan adanya kondisi seperti ini

maka kinerja pemerintah dituntut untuk mampu memenuhi berbagai kebutuhan masyarakat dalam segala aspek. Dalam hal ini pemerintah melalui Dinas Komunikasi dan Informatika (Diskominfo) akan membutuhkan feedback dari masyarakat yang merupakan sebagai sumber informasi untuk meningkatkan kinerja. Pemerintahan yang terbuka didasarkan pada proses kolaboratif dan partisipatif yang mana nantinya akan membentuk sebuah pemerintahan yang bagus, dan oleh karena itu butuh sebuah feedback dari masyarakat [3]. Feedback yang dibutuhkan dari masyarakat tidak hanya yang bersifat positif, melainkan juga bersifat negatif. Masyarakat seringkali memberikan keluhan-keluhan terhadap pelayanan yang telah diberikan atau memberikan usulan-usulan terhadap program kerja atau sekedar ingin mengetahui informasi yang ada saat ini. Salah satu cara untuk mendapatkan feedback dari masyarakat adalah dengan menggunakan sosial media [4]. Berdasarkan data statistik dari sosmedtoday.com hingga Agustus 2014 tentang penggunaan media sosial di Indonesia, terdapat sekitar 70 juta masyarakat Indonesia yang aktif dalam media sosial [5]. Hal ini bisa dibilang bahwa kecenderungan masyarakat cukup tinggi dalam mengakses media sosial yang ada saat ini.

Banyak cara yang disediakan oleh Pemerintah Kota Surabaya menampung partisipasi masyarakat baik dalam bentuk keluhan, informasi, maupun saran pada proses pembangunan kota yang dilaksanakan yaitu via website [www.surabaya.go.id](http://www.surabaya.go.id), facebook sapawarga kota Surabaya, twitter @SapawargaSby, email [mediacenter@surabaya.go.id](mailto:mediacenter@surabaya.go.id), serta portal Kota Surabaya yaitu [sapawarga.surabaya.go.id](http://sapawarga.surabaya.go.id). Selain itu warga Surabaya juga bisa datang langsung ke kantor media center, tepatnya di Jl. Jimerto 6-8 Surabaya [6]. Menurut Kepala Dinas Komunikasi dan Informatika (Diskominfo) Kota Surabaya Antiek Sugiharti agar lebih peka terhadap apa yang terjadi di lapangan,

Pemkot Surabaya mengintegrasikan media center dengan salah satu radio di Surabaya [6].

Data-data berupa keluhan, informasi atau saran yang ada pada media sosial Pemerintah Kota Surabaya dapat dijadikan sebagai sumber data untuk memantau hasil feedback dari masyarakat. Akan tetapi, data-data yang disajikan dari hasil media sosial masih berbentuk terpecah belah sehingga butuh sistem otomatis yang dapat memantau hasil feedback dari masyarakat. Untuk mendapatkan hasil yang dapat dilihat secara jelas maka digunakan proses klasifikasi terhadap opini-opini masyarakat.

Opini masyarakat yang diberikan biasanya bersifat positif, negatif, atau netral. Opini yang diberikan oleh masyarakat inilah yang dapat digunakan sebagai bahan evaluasi oleh pemilik perusahaan atau instansi terkait untuk melakukan perbaikan dan juga dapat meningkatkan kualitas kinerja yang baik ke depannya.

Data-data opini masyarakat bisa kita olah dengan teknik data mining. Pengelompokkan opini-opini bisa kita lakukan dengan teknik data mining yaitu menggali informasi dari sebuah sumber data. Penggalan data biasanya menggunakan 2 metode yaitu klasifikasi dan klustering. Klasifikasi digunakan untuk memprediksi suatu kelas untuk memetakan masing-masing data ke dalam target kelas dengan tujuan memprediksi secara akurat target kelasnya. Klasifikasi yang dilakukan dalam penelitian ini adalah klasifikasi multikelas dimana sistem mempelajari pemetaan dari masukan ke suatu himpunan kelas yang terdiri dari lebih dari dua kelas. Klasifikasi yang akan dilakukan dalam penelitian ini menggunakan metode Naïve Bayes. Metode Naïve Bayes adalah sebuah metode yang digunakan untuk menghitung probabilitas terjadinya suatu peristiwa berdasarkan pengaruh yang didapat dari hasil observasi. Teori ini yang

memungkinkan kita untuk membuat satu model ketidakpastian dari suatu kejadian yang terjadi dengan fakta dari hasil pengamatan.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah dipaparkan di atas, maka rumusan masalah yang akan diselesaikan dalam penelitian ini adalah :

1. Bagaimana cara melakukan pengklasifikasian positif, negatif, dan netral pada komentar masyarakat Kota Surabaya?
2. Bagaimana performa hasil klasifikasi yang telah dilakukan terhadap kinerja pelayanan publik di Kota Surabaya?

### **1.3. Batasan Tugas Akhir**

Adapun batasan masalah yang digunakan pada tugas akhir ini:

1. Data yang digunakan adalah data yang diambil dari twitter @e100ss dan @SapawargaSby sebanyak 5836 data.
2. Data yang diambil adalah data yang berasal dari timeline dan hashtag(#) dari akun @e100ss dan @SapawargaSby.
3. Data hanya diklasifikasikan menjadi 3 yaitu positif, negatif, dan netral.

### **1.4. Tujuan Tugas Akhir**

Tujuan yang hendak dicapai dalam pembuatan tugas akhir ini adalah sebagai berikut

1. Menghasilkan klasifikasi opini masyarakat Surabaya ke dalam 3 kelas yaitu positif, negatif, dan netral berbasis algoritma Naïve Bayes.

2. Mengetahui sentimen masyarakat Surabaya terhadap kinerja Pelayanan Publik di Kota Surabaya
3. Mengetahui tingkat akurasi algoritma dan klasifikasi dalam klasifikasi komentar masyarakat.

### **1.5. Manfaat Tugas Akhir**

Manfaat yang akan didapatkan setelah dilakukan pembuatan tugas akhir adalah sebagai berikut

1. Bagi penulis  
Tugas akhir ini berguna untuk mengetahui kinerja penggunaan algoritma Naïve Bayes dalam sebuah klasifikasi data yang dapat memberikan informasi kepada instansi terkait tentang komentar yang telah diberikan oleh masyarakat Kota Surabaya.
2. Bagi instansi  
Melalui tugas akhir ini instansi terkait dapat memahami sentimen masyarakat terhadap kinerja pelayanan publik yang telah dilakukan dengan melihat klasifikasi komentar masyarakat kota Surabaya.

### **1.6. Relevansi**

Relevansi tugas akhir ini terhadap laboratorium Akuisisi Data dan Diseminasi Informasi adalah karena tugas akhir ini berkaitan dengan penerapan mata kuliah bidang keilmuan laboratorium ADDI. Mata kuliah tersebut antara lain Sistem Cerdas, Sistem Pendukung Keputusan, dan Penggalan Data dan Analitika Bisnis khususnya bagian klasifikasi.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dijelaskan mengenai teori-teori yang mendasari penelitian dan mengenai penelitian terkait dengan penelitian ini. Teori yang dijelaskan antara lain mengenai penggalian teks, metode praproses data teks, metode klasifikasi teks, dan pengukuran performa klasifikasi. Sedangkan penelitian-penelitian terkait yang dikaji antara lain :

#### **2.1. Penelitian Sebelumnya**

Tugas akhir ini memiliki konteks dasar metode klasifikasi sentimen twitter. Dalam pengerjaan tugas akhir ini, akan digunakan beberapa penelitian sebelumnya sebagai pedoman dan referensi pengerjaan. Pada tabel di bawah ini akan dijelaskan deskripsi, metode, datayang digunakan dan hasilnya. Dalam mengerjakan tugas akhir ini terdapat penelitian terkait yang sudah dilakukan seperti “Ekstraksi Informasi Transaksi Online pada Twitter” oleh Masayu Leylia Khodra dan Ayu Purwarianti dari Institut Teknologi Bandung. Paper ini mengulas mengenai analisis konten tweet untuk mendapatkan informasi transaksi online di Indoensia dengan mengembangkan aplikasi SaFE-F yaitu aplikasi yang melakukan pencarian (*Search*) dan Filter tweet yang relevan, Ekstraksi informasi transaksi online, dan menyimpan hasil ekstraksinya (*Filling*). Paper ini bertujuan mengekstraksi informasi transaksi online di Indonesia dengan melakukan klasifikasi tweet dan ekstraksi informasi.



Tabel 2.1 Ringkasan Studi Sebelumnya

No.	Judul	Penulis	Metode yang digunakan	Data yang digunakan	Hasil
1.	Ekstraksi Informasi Transaksi Online pada Twitter	Masayu Leylia Khodra dan Ayu Purwarianti	Klasifikasi tweet dengan menggunakan algoritma C4.5, fitur trigram, dan tanpa pra-proses. Sedangkan untuk klasifikasi token menggunakan algoritma pembelajaran Ibk dan 7 set fitur terbaik gain ratio	5000 tweet, 1875 tidak relevan, 369 tweet relevan, dan 2756 tweet berkategori iklan	Pada tahapan filter didapatkan akurasi sebesar 85,09 dengan menggunakan algoritma C4.5 dan fitur trigram. Untuk hasil ekstraksi didapatkan akurasi sebesar 81,49% dengan menggunakan algoritma pembelajaran Ibk dan 7 set fitur terbaik gain ratio

2.	Memfaatkan Big Data Untuk Mendeteksi Emosi	Aprianti Putri Sujana	2.488.982 tweet	LIBLINEAR , Multinomial Naive Bayer(MNB)	Kinerja LIBLINEAR classifier pada masing-masing kategori emosi. Terdapat tiga emosi yang paling populer joy, sadness, dan anger
3.	Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik	Ismail Sunni dan Dwi Hendratmo Widyantoro	Untuk analisis sentimen menggunakan algoritma Naive Bayes dan untuk melihat karakter topik yang muncul menggunakan metode Tf-Idf	2000 tweet yang dilabeli sebagai positif, negatif, netral	Dengan metode Naive Bayes dan menggabungkan 10 metode pra-proses, sehingga akurasi didapatkan antara 69,4-72,8%

4.	Twitter as a Corpus for Sentiment Analysis and Opinion Mining	Alexander Pak, Patrick Paroubek	Menggunakan metode Naive Bayes, dengan mencoba melakukannya dengan metode lain yaitu SVM dan CRF	216 tweet dari akun twitter agensi koran dan majalah di US yang dibagi ke dalam 3 kelas yaitu mengandung emosi positif, negatif, dan tidak menunjukkan emosi apapun	Dengan menggunakan unigram, bigram, dan trigram nilai akurasi berada pada nilai $\pm 80\%$
5.	Twitter Sentimen	Alec Go, Richa	Menggunakan 3 metode yaitu	Dataset yang digunakan	Dengan menggunakan:

	t Classifica tion using Distant Supervisi on	Bhayani, dan Lei Huang	Naive Bayes, SVM, dan Maximum Entropi. Ekstraksi fitur dilakukan dengan unigram, bigram, unigram dengan bigram, dan unigram dengan POS tag	sebanyak 359 yaitu 177 negatif dan 182 positif	fitur unigram, SVM mempunyai akurasi terbesar yaitu 82,2% fitur bigram, Naive Bayes mempunyai akurasi terbesar yaitu 81,6% fitur unigram+bigram Maximum Entropi mempunyai akurasi terbesar yaitu 83% fitur unigram+POS SVM mempunyai akurasi terbesar yaitu 81,9%
--	---	------------------------------	---	--	--

Penelitian terkait yang lain yaitu “Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik” oleh Ismail Sunni dan Dwi Hendratmo Widyantoro dari Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika. Jurnal ini mengambil topik tentang analisis sentimen dengan menggunakan data dari jejaring sosial seperti Twitter. Pemanfaatan data dari Twitter digunakan untuk mengetahui opini atau sentimen pengguna jejaring sosial terhadap suatu topik. Salah satu topik yang penting adalah tokoh publik, seperti calon gubernur, ketua partai, dan lain sebagainya yang akan dijadikan sebagai domain dalam jurnal ini. Permasalahan utama yang muncul dalam jurnal ini adalah model bahasa pengguna Twitter di Indonesia yang memiliki karakteristik yang berbeda dari pengguna di luar Indonesia. Ismail Sunni dan Dwi Hendratmo menawarkan solusi yaitu F3(F3 is Factor Finder) yang memiliki beberapa metode praproses yang diperkirakan mampu menangani permasalahan model bahasa yang ditemukan. F3 menggunakan metode algoritma Naïve Bayes untuk melakukan analisis sentimen. Sedangkan untuk mengetahui perubahan sentimen, akan menampilkan perubahan sentimen dalam bentuk kurva. Dan untuk mengetahui topik yang menyebabkan perubahan sentimen, digunakan metode Tf-Idf dengan discounted-cumulative untuk menangani karakter topik yang muncul di Twitter yang berkelanjutan.

Menurut Alexander Pak, Patrick Paroubek dari Universitas de Paris-Sud, Laboratoire LIMSI-CNRS dengan paper “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, paper ini menunjukkan bagaimana untuk secara otomatis mengumpulkan korpus untuk tujuan analisis sentimen dan pertambangan pendapat. Fokusannya yaitu menggunakan

twitter yang merupakan salah satu platform microblogging yang populer untuk melakukan analisis sentimen. Pada paper ini ditunjukkan bagaimana melakukan analisis linguistik dari korpus yang dikumpulkan dan menjelaskan fenomena yang ditemukan. Dengan menggunakan korpus, penulis mencoba membangun classifier sentimen yang mana mampu untuk menentukan sentimen positif, negatif, dan netral untuk sebuah dokumen. Pada paper ini menggunakan Tree Tagger untuk melakukan proses POS tagging dan mengamati perbedaan distribusinya ke dalam 3 bagian yaitu positif, negatif, dan netral. Kesimpulan dari paper ini adalah penggunaan strategi filtering dan common n-grams(salience dan entropy) lebih akurat dengan salience daripada yang menggunakan entropy. Ketika ingin meningkatkan ukuran sampel maka kita juga harus meningkatkan kinerja sistem dan pada titik tertentu ketika dataset cukup besar maka proses perbaikan tidak dapat dicapai hanya dengan meningkatkan ukuran data pelatihan.

Satu paper terakhir yaitu “Twitter Sentiment Classification using Distant Supervision” oleh Alec Go, Richa Bhayani, dan Lei Huang dari Stanford University. Paper ini mengelompokkan sentimen dari pesan(message) twitter. Pesan-pesan ini diklasifikasikan positif dan negatif. Klasifikasi ini menggunakan metode SVM, Naïve Bayes, dan Maksimum Entropi. Untuk mendapatkan data-data tersebut menggunakan bantuan Twitter API yang dengan mudah digunakan untuk mengambil data dalam jumlah yang besar. Dalam paper ini juga dilakukan langkah-langkah dalam preprocessing data untuk menghasilkan akurasi yang tinggi yaitu dengan mendefinisikan sentimen yang ada termasuk ke dalam klasifikasi yang mana apakah positif, negatif, dan netral. Selanjutnya melakukan karakteristik tweet yang dilihat dari

panjangnya, ketersediaan data, model bahasanya, dan domain pengguna twitter.

## **2.2. Dasar Teori**

Pada sub bab dasar teori akan dijelaskan tentang teori-teori mengenai Twitter, Klasifikasi, Model Klasifikasi, Data Mining, Text Corpus, Praproses Data, Algoritma Naïve Bayes, Algoritma SVM, dan Evaluasi Performa Hasil Klasifikasi.

### **2.2.1. Twitter**

Twitter adalah salah satu layanan jejaring sosial yang dioperasikan oleh Twitter, Inc. Twitter didirikan pada bulan Maret 2006 oleh Jack Dorsey, dan situs jejaring sosialnya diluncurkan pada bulan Juli. Sejak diluncurkan, Twitter telah menjadi salah satu dari sepuluh situs yang paling sering dikunjungi di Internet, dan dijuluki dengan "pesan singkat dari Internet". Twitter menawarkan jaringan sosial berupa *microblog*. Disebut *microblog* karena situs ini memungkinkan penggunaanya mengirim dan membaca pesan blog seperti pada umumnya namun terbatas hanya sejumlah 140 karakter yang ditampilkan pada halaman profil pengguna. Twitter memiliki karakteristik dan format penulisan yang unik dengan simbol ataupun aturan khusus. Pesan dalam Twitter dikenal dengan sebutan *tweet* [7].

#### **2.2.1.1. @SapawargaSby**

Akun twitter SapawargaSby dibuat sejak April 2011. Akun ini dibuat oleh Dinas Komunikasi dan Informatika Kota Surabaya dalam menjangking keluhan/pengaduan masyarakat melalui jejaring media sosial [8]. Selain itu, akun twitter SapawargaSby juga bisa digunakan masyarakat dalam menyampaikan aspirasinya tentang kinerja pemerintah Kota Surabaya.



### **2.2.1.2. @e100ss**

Akun twitter @e100ss adalah akun resmi dari radio Suara Suraba 100 FM. Radio Suara Surabaya ini mengklaim sebagai radio pertama di Indonesia yang menerapkan format radio news interaktif solutif atau informasi [9]. Akun ini dibuat sejak September 2012. Akun twitter @e100ss dapat memberikan informasi terbaru dengan cepat kepada para followersnya. Akun twitter @e100ss memiliki keunikan jika dibandingkan dengan akun twitter jasa informasi lainnya, keunikannya adalah pihak @e100ss yang memberikan informasi, followers-nya pun juga dapat berpartisipasi dalam pemberian informasi seputar keadaan lalu lintas di Jawa Timur khususnya kota Surabaya dengan mengirim info dengan cara menulis di twitter sendiri lalu dengan menambahkan mention ke @e100ss. Dengan keunikan ini sering terjadi feedback yang baik antara followers-nya yang satu dengan yang lainnya, yang kemudian seringkali terjadi pertukaran informasi di dalamnya [10].

### **2.2.2. Klasifikasi**

Secara harfiah arti dari klasifikasi yaitu penggolongan atau pengelompokkan. Ada beberapa pengertian mengenai klasifikasi, menurut KBBI klasifikasi adalah penyusunan bersistem di kelompok atau golongan menurut kaidah atau standar yang ditetapkan [11]. Sedangkan menurut Olson dan Shi (2013), klasifikasi merupakan metode yang ditunjukkan untuk pembelajaran fungsi-fungsi berbeda yang memetakan masing-masing data terpilih ke dalam salah satu dari kelompok kelas yang telah ditetapkan sebelumnya. Menurut Han dan Kamber (2011) [12], klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau

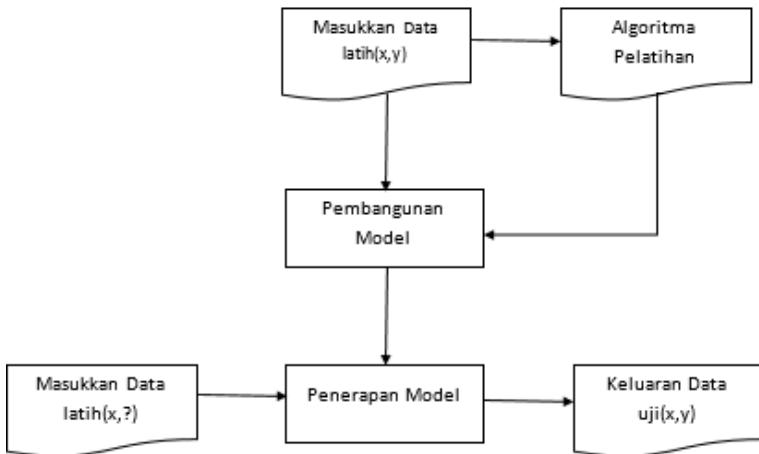
membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Sehingga dapat diambil kesimpulan bahwa klasifikasi adalah sebuah teknik dalam data mining yang dilakukan untuk memprediksi suatu kelas untuk memetakan masing-masing data ke dalam suatu kategori atau target kelas dengan tujuan memprediksi secara akurat kelas target untuk setiap kasus dalam data yang digunakan.

#### **2.2.2.1. Klasifikasi Teks**

Klasifikasi teks adalah teknik dalam melakukan pengelompokan sebuah kategori subjek, topik atau suatu genre [13]. Definisi dari klasifikasi teks yaitu diberikan sebuah masukan berupa dokumen  $d$ , terdapat sebuah kelas yang sudah tetap yaitu  $C = \{c_1, c_2, \dots, c_3\}$ . Dan nantinya keluaran yang dihasilkan yaitu sebuah prediksi kelas  $c \in C$ . Contoh dari klasifikasi teks adalah deteksi spam, identifikasi penulis/pengarang sebuah buku dan sejenisnya, identifikasi usia atau gender, identifikasi bahasa, analisis sentimen, dan lain-lain. Dalam klasifikasi teks, ada beberapa metode yang biasa digunakan yaitu Naïve Bayes, Hand Coded Rules, dan Support Vector Machine.

#### **2.2.3. Model Klasifikasi**

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, dimana ada suatu model yang menerima masukan kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban atau sebuah hasil. Kerangka kerja klasifikasi bisa kita lihat pada gambar di bawah ini.



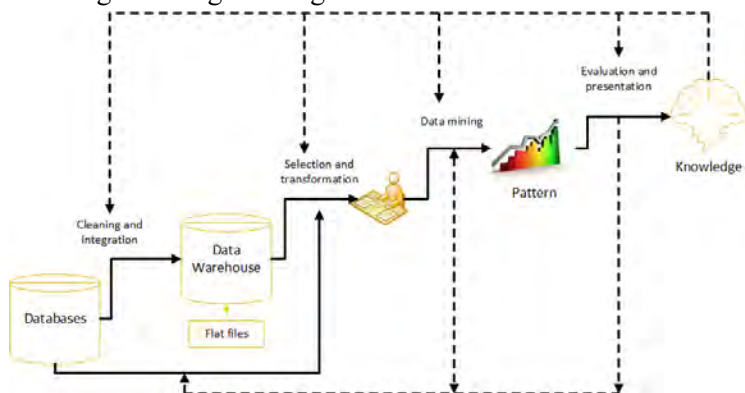
**Gambar 2.1 Model Klasifikasi**

Disediakan sejumlah data  $latih(x,y)$  untuk digunakan sebagai data pembangunan model. Model tersebut kemudian dipakai untuk memprediksi kelas dari data  $uji(x,y)$  sehingga diketahui kelas  $y$  yang sesungguhnya [14]. Model yang sudah dibangun pada saat pelatihan kemudian dapat digunakan memprediksi label kelas baru yang belum diketahui. Dalam pembangunan model selama proses pelatihan tersebut diperlukan suatu algoritma untuk membangunnya, yang disebut algoritma pelatihan. Dalam penelitian ini akan menggunakan algoritma Naïve Bayes.

#### **2.2.4. Data Mining**

Menurut Han and Kamber(2000) data mining adalah sebuah proses untuk menemukan pengetahuan yang menarik dari data dalam jumlah yang besar [15]. Data mining adalah sebuah teknologi yang memadukan metode analisis data dengan algoritma yang canggih untuk memproses data dalam volume

yang besar [16]. Data mining merujuk pada ekstraksi atau pengetahuan pertambangan dari data dalam jumlah besar. Data mining seharusnya lebih tepat bernama "pertambangan pengetahuan dari data" yang sayangnya agak panjang. "pengetahuan pertambangan" istilah yang lebih pendek, mungkin tidak mencerminkan penekanan pada pertambangan dari sejumlah besar data. Namun demikian, pertambangan adalah istilah yang jelas karakteristik proses yang menemukan satu set kecil nugget berharga dari banyak bahan baku. Selain itu, banyak hal lain memiliki makna yang sama dengan data mining misalnya, pertambangan pengetahuan dari data, ekstraksi pengetahuan, analisis data/pola, data arkeologi, dan data pengerukan [17]. Proses penemuan pengetahuan ditunjukkan dalam gambar di bawah ini sebagai urutan berulang dari langkah-langkah berikut:



**Gambar 2.2 Proses Data Mining**

**(Sumber : Data Mining Concepts and Techniques, J. Han, M. Kamber dan J. Pei)**

Penjelasan :

1. Data cleaning : untuk menghilangkan noise dan data yang tidak konsisten.
2. Data integration : mengkombinasikan atau mengintegrasikan beberapa sumber data.
3. Data selection : mengambil data-data yang relevan dari database untuk dianalisis.
4. Data transformation : mentransformasikan data summary ataupun operasi agregasi
5. Data mining : merupakan proses yang esensial dimana metode digunakan untuk mengekstrak pola data yang tersembunyi.
6. Pattern evaluation : untuk mengidentifikasi pola sehingga merepresentasikan pengetahuan berdasarkan nilai-nilai yang menarik.
7. Knowledge presentation : dimana teknik representasi dan visualisasi data digunakan untuk mempresentasikan pengetahuan yang didapat kepada user.

### **2.2.5. Text Corpus**

Korpus adalah kumpulan kata yang mendeskripsikan penggunaan kata dalam suatu dokumen. Linguistik korpus adalah metode linguistik yang menggunakan data dari bahan-bahan bahasa yang terkumpul dalam suatu sumber yang disebut korpus atau korpora(sejenis "bank" bahasa) yang berasal dari penggunaan bahasa dalam berbagai genre, ragam, dan bahan lisan maupun tertulis [18]. Kamus kata yang dihasilkan dari text corpus ini nantinya akan digunakan dalam proses pengerjaan selanjutnya yaitu pada tahapan praproses data.

### 2.2.6. Praproses data

Tahapan praproses data dilakukan dengan mengolah data mentah yang kita peroleh. Data mentah tersebut diolah dengan mengikuti proses-prosesnya agar dihasilkan data yang siap digunakan dalam penelitian [19] [20]. Tahapan proses yang dilakukan antara lain :

- a. Cleansing, yaitu proses membersihkan dokumen dari kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah karakter HTML, kata kunci, ikon emosi, *hashtag*(#), *username*(@username), *url*(http://situs.com), dan email(nama@situs.com).
- b. Case folding, yaitu penyeragaman bentuk huruf serta penghapusan angka dan tanda baca. Dalam hal ini yang digunakan hanya huruf latin antara a sampai dengan z.
- c. Tokenizing, yaitu proses pengubahan dari kalimat menjadi kata-kata.
- d. Stopword removal, merupakan proses pemilahan kata penghubung seperti “dan”, “pada”, dan lain sebagainya.
- e. Stemming, merupakan proses mengambil kata dasar dari sebuah kata yang memiliki imbuhan, misal “mengapresiasi” menjadi “apresiasi”.

### 2.2.7. Algoritma Naive Bayes

Algoritma Naïve Bayes merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. Naïve Bayes menggunakan asumsi bahwa dalam sebuah dokumen kemunculan kata tidak mempengaruhi kemunculan kata yang lain dan ketidakmunculan kata tidak mempengaruhi ketidakmunculan kata yang lain. Metode ini ditemukan pada tahun 1702 oleh Thomas Bayes. Algoritma ini dikenal sebagai Teorema Bayes pada masa lampau yang merupakan pengklasifikasian dengan

metode probabilitas dan statistik [21]. Dalam sebuah kemungkinan adalah mungkin bahwa X akan terjadi dengan diberikan bukti Y yang biasa dituliskan dengan  $P(X | Y)$ . Aturan Bayes memungkinkan kita untuk menentukan probabilitas ini ketika semua yang kita miliki adalah probabilitas hasil yang berlawanan dan dari dua komponen individual:  $P(X | Y) = P(X) P(Y | X) / P(Y)$ . Pernyataan ini bisa sangat membantu ketika kita mencoba untuk memperkirakan probabilitas sesuatu berdasarkan contoh itu terjadi dengan rumus di bawah ini [22]:

$$P(\text{Sentimen} | \text{Sentence}) = \frac{P(\text{Sentimen})P(\text{Sentence} | \text{Sentiment})}{P(\text{Sentence})} \quad (1)$$

Metode Naïve Bayes terdiri dari 2 fase tahapan yaitu fase tahap pembelajaran dan fase tahap klasifikasi. Dalam fase pembelajaran adalah fase dimana dokumen hasil praproses melewati proses pembelajaran untuk didapatkan data pelatihan. Fase ini digunakan untuk mendapatkan nilai probabilitas dari  $P(V_j)$  dan  $P(W_k|V_j)$ . Fase pembelajaran ini dimulai dari masukan yang terdiri dari data yang kita ambil kemudian hingga didapatkan kata-kata unik yang mencerminkan masing-masing kelas klasifikasi yang telah diidentifikasi. Kata-kata yang sudah terdefinisi inilah yang akan digunakan pada tahapan selanjutnya yaitu fase klasifikasi. Berikut adalah rumus yang akan digunakan untuk menghitung perbandingan dari jumlah kelas dokumen yang ada  $P(V_j)$  dalam setiap kategori :

$$P(V_j) = \frac{|fd(V_j)|}{|D|} \quad (2)$$

Dimana  $fd(V_j)$  adalah jumlah kata-kata di setiap kategori j, dan  $|D|$  adalah jumlah kelas dokumen untuk dijadikan model. Selanjutnya dilakukan proses menghitung indeks kelas

dengan melibatkan frekuensi kata kunci dokumen dituliskan dengan rumus dibawah ini :

$$P(W_k|V_j) = \frac{f(W_k|V_j)+1}{N+|W|} \quad (3)$$

Dimana  $P(W_k|V_j)$  adalah jumlah kejadian dari kata  $W_k$  dalam kategori  $V_j$ . Sedangkan  $N$  adalah jumlah semua kata yang ada pada kategori  $V_j$ , dan  $|W|$  adalah jumlah kata unik dalam data pelatihan.

Selanjutnya akan dilakukan tahap klasifikasi, pada tahapan ini dokumen juga melalui praproses data terlebih dahulu. Proses ini dilakukan berdasarkan data pelatihan yang telah dihasilkan oleh fase pembelajaran. Proses perhitungan digunakan untuk menghitung nilai tiap-tiap klasifikasinya yang dimulai dengan memasukkan dokumen baru ke perhitungan Naïve Bayes. Selanjutnya hasil dari perhitungan yang telah ada dipilih nilai yang terbesar sebagai hasil klasifikasi [23].

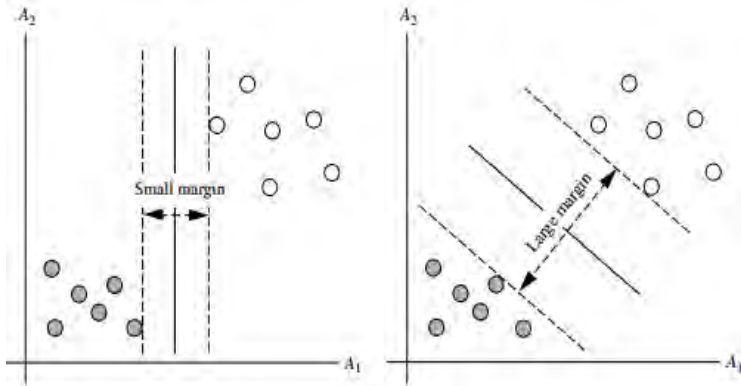
$$V_{map} = \underset{V_j \in V}{\operatorname{argmax}} P(V_j) \prod_i P(W_k|V_j) \quad (4)$$

### 2.2.8. Support Vector Machine(SVM)

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition [24]. SVM adalah sebuah algoritma yang bekerja sebagai berikut, menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi. Dalam dimensi baru ini, akan mencari linear hyperplane pemisah optimal (yaitu, "batas keputusan" memisahkan tupel dari satu kelas dari yang lain). Dengan pemetaan nonlinear yang tepat untuk dimensi yang cukup tinggi, data dari dua kelas selalu dapat dipisahkan dengan hyperplane. SVM menemukan hyperplane



ini menggunakan vektor dukungan ("penting" tupel pelatihan) dan margin (didefinisikan oleh vektor dukungan) [17].



**Gambar 2.3 Support Vector pada Metode SVM**

(Sumber : Data Mining Concept and Techniques, 2011)

Mendapatkan ke definisi informal margin, kita dapat mengatakan bahwa jarak terpendek dari hyperplane ke satu sisi margin adalah sama dengan jarak terpendek dari hyperplane ke sisi lain dari margin, di mana "sisi" dari margin yang sejajar dengan hyperplane tersebut. Support vector machine (SVM) adalah suatu teknik yang relatif baru untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi, yang sangat populer belakangan ini.

SVM berada dalam satu kelas dengan ANN dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas supervised learning [25]. Pada *machine learning*, terdapat istilah *kernel trick* yang merupakan metode yang menggunakan algoritma *linier classifier* untuk menyelesaikan permasalahan nonlinier dengan cara memetakan dimensi input ke ruang dimensi yang lebih tinggi,

sehingga membuat *linier classifier* di ruang dimensi yang baru setara dengan *non linear classifier* di ruang dimensi asli. Dengan *kernel*, fungsi pemetaan tidak pernah dihitung secara eksplisit, karena ruang dimensi tinggi yang digunakan memungkinkan pada dimensi yang tak terbatas [26]. Berikut adalah beberapa fungsi kernel yang digunakan [27]:

- Linear  

$$K(x_i, x_j) = x_i^T x_j$$
- Polynomial  

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$$
- Radial Basis Function(RBF)  

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$$
- Sigmoid  

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$$

### 2.2.9. Evaluasi performa klasifikasi

Dalam klasifikasi teks juga perlu mengetahui apakah penggunaan metode sudah tepat atau belum. Klasifikasi yang dilakukan dalam penelitian ini termasuk dalam klasifikasi multikelas yaitu terdiri dari klasifikasi positif, negatif, dan netral. Pada klasifikasi multikelas, akurasi hanya mengukur performa pengklasifikasi secara global, yaitu pada seluruh kelas yang ada. Ada beberapa pengujian yang dilakukan untuk mengevaluasi keefektifan metode yaitu dengan cara menghitung akurasi multikelas, serta *Precision* dan *Recall*. Selain itu digunakan juga parameter pengukuran yang baru yaitu *F-Measure*.

#### 2.2.9.1. Akurasi

Menghitung akurasi yang dimaksudkan yaitu untuk menghitung ketepatan prediksi dari suatu pengklasifikasi dan bisa kita lihat dalam confusion matrix. Misalkan pada suatu

klasifikasi terdapat kelas A,B, dan C. Berikut adalah confusion matrix yang dapat dibuat ketika terdapat 3 kelas yang akan dipakai [28].

	Prediksi c = A	Prediksi c = B	Prediksi c = C	
Actual c = A	True A	(FP B   c = A)	(FP C   c = C)	Total Actual c = A
Actual c = B	(FP A   c = B)	True B	(FP C   c = B)	Total Actual c = B
Actual c = C	(FP A   c = C)	(FP B   c = C)	True C	Total Actual c = C
	Total Prediksi c = A	Total Prediksi c = B	Total Prediksi c = C	TOTAL

**Gambar 2.4 Confusion matrix**

(Sumber : Peter Flach, 2012)

Cara menghitung akurasi pada suatu klasifikasi yaitu :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah klasifikasi benar}}{\text{Jumlah total data}} \\
 &= \frac{\text{True A} + \text{True B} + \text{True C}}{\text{Jumlah total}}
 \end{aligned}
 \tag{5}$$

### 2.2.9.2. Precision, Recall, dan F-Measure

*Precision* digunakan untuk mengukur ketepatan prediksi pengklasifikasi pada kelas tertentu dan *Recall* digunakan untuk mengukur berapa banyak instans dari kelas tertentu yang dapat diprediksikan secara benar [28]. Pada suatu kelas klasifikasi terdapat kelas A, B, dan C, maka bisa dihitung bahwa nilai *precision* dan *recall* dari kelas A yaitu dengan rumus di bawah ini :

$$\text{Precision}(A) = \frac{TP}{TP+FP} \tag{6}$$

$$\text{Recall}(A) = \frac{TP}{TP+FN} \tag{7}$$

Berikut adalah penjelasan dari rumus *precision* dan *recall* yang terdiri dari TP, TN, FP, dan FN :

- True Positives (TP)  
Titik data yang diklasifikasikan model sebagai positif, dan label sesungguhnya memang positif(prediksi betul).
- True Negatives (TN)  
Titik data yang diklasifikasikan model sebagai negatif, dan label sesungguhnya memang negatif(prediksi betul).
- False Positives (FP)  
Titik data yang diklasifikasikan model sebagai positif, dan label sesungguhnya adalah negatif(prediksi salah).
- False Negatives (FN)  
Titik data yang diklasifikasikan model sebagai negatif, dan label sesungguhnya adalah positif(prediksi salah).

### 2.2.9.3. F-Measure

Jika hanya menggunakan *precision* dan *recall* untuk parameter evaluasi *information retrieval*, hasil yang didapatkan tidak akan optimal karena:

- Nilai *precision* dan *recall* mengandung *trade-off*
- Setiap pengguna mempunyai kebutuhan berbeda antara *precision* dan *recall*.

Untuk itu, pada penelitian ini menggunakan parameter pengukuran yang baru, yaitu *F-Measure* yaitu pengukuran yang mengkombinasikan *precision* dan *recall* yang diterapkan ke dalam deret harmonik [29] [28]. Berikut adalah rumus umum untuk menghitung F-measure:

$$F = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

## **BAB III**

### **METODE PENELITIAN**

Dalam melakukan penelitian tugas akhir ini, ada beberapa tahapan yang harus dilakukan. Tahap-tahap tersebut memiliki detail penjelasan dan jangka waktu masing-masing yang telah ditentukan untuk membantu penulis dalam menyelesaikan pengerjaan tugas akhir ini. Pada gambar 3.1 akan dijelaskan tentang keluaran dari masing-masing tahap yang terlihat pada bagian kanan dari alur pengerjaan.

#### **3.1. Studi Literatur**

Studi literatur yang dilakukan dalam pembuatan tugas akhir ini adalah pembelajaran literatur terkait dengan konsep serta metode yang digunakan untuk menyelesaikan permasalahan dalam tugas akhir ini, seperti pembelajaran mengenai pola data, metode algoritma yang mendukung, dan sebagainya. Sumber studi literatur yang digunakan dalam tugas akhir ini adalah :

- E-book  
Ebook digunakan untuk memperdalam teori penulis dalam mengerjakan tugas akhir. E-book yang digunakan meliputi teori klasifikasi dengan Naïve Bayes dan teknik dalam melakukan data mining.
- Jurnal ilmiah  
Jurnal ilmiah digunakan untuk mempelajari penelitian terdahulu mengenai tahapan melakukan klasifikasi teks yang bisa dijadikan sebagai acuan pemilihan metode dan juga sebagai sebuah perbandingan. Untuk hasil dari studi literatur jurnal ilmiah telah disajikan pada Tabel 2.1.
- Artikel

Artikel ini digunakan untuk menemukan fakta-fakta yang ada saat ini untuk mendukung latar belakang permasalahan. Artikel yang digunakan dalam tugas akhir ini adalah artikel berita dan yang terkait dengan kinerja pemerintahan.

### **3.2. Pengumpulan Data**

Pada tahapan ini terdapat beberapa data yang harus disiapkan. Data-data tersebut merupakan data utama yang akan digunakan di dalam penelitian ini. Adapaun data-data yang harus disiapkan adalah sebagai berikut :

1. Data Tweets @e100ss

Tahap pengumpulan data tweet @e100ss dilakukan dengan proses crawling menggunakan sebuah aplikasi crawler. Dengan menggunakan aplikasi ini kita bisa mengambil seluruh data tweet yang kita butuhkan dan dapat kita gunakan sesuai dengan kebutuhan analisis kita.

2. Data Tweets @SapawargaSby

Tahap pengumpulan data tweet @SapawargaSby dilakukan dengan proses crawling menggunakan sebuah aplikasi crawler. Dengan menggunakan aplikasi ini kita bisa mengambil seluruh data tweet yang kita butuhkan dan dapat kita gunakan sesuai dengan kebutuhan analisis kita.



### **3.3. Pemrosesan Data**

Dalam tahapan pemrosesan data akan dilakukan 2 proses di dalamnya yaitu tahap pelabelan data yaitu pelabelan kelas klasifikasi dan praproses data.

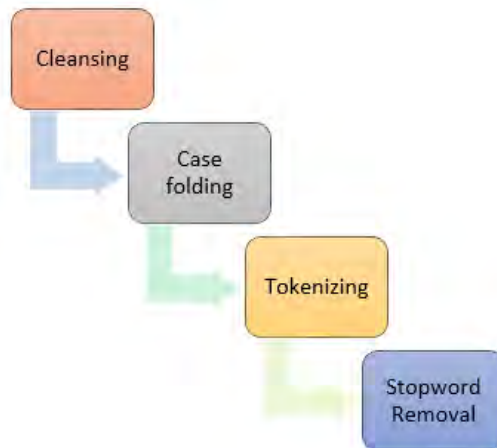
#### **3.3.1. Tahap Pelabelan Data**

Data yang digunakan dari @e100ss dan @SapawargaSby memiliki kecenderungan positif, negatif, dan netral. Data-data tersebut dilakukan proses pemilahan untuk memastikan proses klasifikasi awal yaitu mengelompokkan atau memasukkan data teks ke dalam suatu kelas. Data-data yang telah diambil akan berpengaruh terhadap perhitungan akurasi yang dilakukan dengan metode Naïve Bayes dan Support Vector Machine (SVM).

#### **3.3.2. Tahap Praproses Data**

Pada tahapan ini data mentah yang diperoleh dilakukan pemrosesan untuk mengubahnya menjadi data yang siap digunakan. Adapun proses-proses yang dilakukan dalam tahapan ini adalah membersihkan dokumen dari kata yang tidak diperlukan untuk mengurangi noise, menyeragamkan bentuk huruf serta penghapusan angka dan tanda baca, mengubah dari kalimat menjadi kata-kata, melakukan proses pemilahan kata yang merupakan kata penghubung dan mengambil kata dasar dari sebuah kata yang memiliki imbuhan. Untuk tahap praproses data ini, semua langkah-langkahnya dilakukan secara manual. Berikut adalah alur praproses data yang akan dilakukan dalam penelitian ini :





**Gambar 3.2 Tahap praproses data**

Penjelasan :

1. Cleansing

Cleansing yaitu proses membersihkan dokumen dari kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah karakter HTML, kata kunci, ikon emosi, *hashtag*(#), *username*(@username), *url*(http://situs.com), dan email(nama@situs.com).

2. Case folding

Case folding yaitu penyeragaman bentuk huruf serta penghapusan angka dan tanda baca. Dalam hal ini yang digunakan hanya huruf latin antara a sampai dengan z.

3. Tokenizing

Tokenizing yaitu proses pengubahan dari kalimat menjadi kata-kata.

4. Stopword removal

Stopword removal merupakan proses pemilahan kata penghubung seperti “dan”, “pada”, dan lain sebagainya.

### **3.4. Uji Coba dan Proses Klasifikasi**

Pada tahapan ini dilakukan uji algoritma Naïve Bayes dan juga evaluasi model atau evaluasi performa dari algoritma yang digunakan dalam penelitian ini. Dalam tahapan uji coba digunakan tools R untuk mengerjakan semua prosesnya.

#### **3.4.1. Tahap pelatihan dengan algoritma Naïve Bayes**

Pada tahap ini akan dilakukan tahap latih terhadap opini masyarakat yang sudah diketahui klasifikasinya, sehingga didapatkan model probabilitas algoritma Naïve Bayes yang nantinya akan digunakan pada tahap uji.

#### **3.4.2. Evaluasi performa klasifikasi**

Pada tahap ini akan dilakukan perhitungan evaluasi performa algoritma yang telah dilakukan pada saat tahap uji yaitu dengan menghitung akurasi, precision dan recallnya. Tahapan ini bertujuan untuk mengetahui keakurasian model klasifikasi yang diklasifikasikan menjadi 3 kategori yang telah diberikan di awal. Jika pada hasil evaluasi terdapat kekurangan maka dilakukan proses perbaikan pada pembuatan model klasifikasi dan dilakukan evaluasi lagi. Jika hasil sudah baik langkah selanjutnya adalah melakukan analisis terhadap hasil yang diperoleh.

### **3.5. Pembahasan hasil percobaan**

Pada tahap ini akan dilakukan pembahasan dari semua hasil percobaan model yang telah dilakukan pada proses klasifikasi dengan menggunakan algoritma Naïve Bayes. Adapun dalam

pengerjaan model klasifikasi tidak hanya menggunakan Naïve Bayes melainkan menggunakan algoritma SVM. Untuk pembahasan lebih lanjut akan dijelaskan pada bab 5.

### **3.6. Visualisasi hasil**

Pada tahap ini akan dilakukan perancangan dan implementasi visualisasi. Visualisasi yang dilakukan pada tugas akhir ini yaitu dengan menggunakan tools R. Visualisasi yang akan dikerjakan terdapat 2 hal yaitu pembuatan worcloud dan streamgraph.

### **3.7. Pembuatan Buku Laporan**

Pada tahapan ini merupakan tahap terakhir yang dilakukan yaitu pembuatan laporan tugas akhir yang merupakan dokumentasi dari seluruh pengerjaan yang telah dilakukan. Tahapan ini dilakukan setelah di dapatkan hasil klasifikasi yang terbaik beserta dengan analisis dan pembahasannya. Laporan tugas akhir ini selain bertujuan sebagai dokumentasi pengerjaan dari awal hingga akhir adalah sebagai referensi untuk pengerjaan penelitian selanjutnya dan dapat dikembangkan lebih lanjut dengan topik tugas akhir yang serupa.

*Halaman ini sengaja dikosongkan*

## **BAB IV**

### **PERANCANGAN**

Pada bab ini akan dijelaskan bagaimana rancangan dari penelitian tugas akhir yang meliputi subyek dan obyek penelitian, pemilihan subyek dan obyek penelitian dan bagaimana penelitian akan dilakukan

#### **4.1. Pengumpulan Data**

Pengumpulan data adalah proses yang digunakan dalam pengerjaan tugas akhir ini. Pengumpulan data merupakan hal yang mutlak harus dilakukan untuk penelitian ini karena seluruh proses klasifikasi berdasarkan pada data ini. Proses pengumpulan data dilakukan dengan melakukan teknik crawling pada twitter. Data yang digunakan diambil dari 2 akun twitter yaitu @e100ss dan @SapawargaSby. Data mentah yang didapatkan dari hasil crawling [30] diunduh dalam bentuk file .xls dan didapatkan atribut-atribut sebagai berikut yang terlihat pada tabel :

**Tabel 4.1 Atribut data**

<b>Atribut</b>	<b>Tipe data</b>	<b>Keterangan</b>
Id_message	Varchar	Urutan data tweet
Text_message	Text	Merupakan tweet-tweet yang telah diambil
Date_message	Date	Waktu pengambilan tweet
Id_kategori	Int	-

Account	Varchar	Akun orang yang mengtweet
Lattitude	Double	Lokasi orang yang mengtweet
Longitude	Double	Lokasi orang yang mengtweet

Berdasarkan atribut di atas maka didapatkan keseluruhan data yang berjumlah 5836 data dalam rentang waktu dari 1 September 2015 - 13 Oktober 2015. Tabel 4.2 adalah beberapa data komentar yang telah didapatkan.

#### 4.2. Penyeleksian Atribut

Setelah mendapatkan data dari tahap pengumpulan data maka selanjutnya adalah tahapan penyeleksian atribut. Penyeleksian atribut disini yaitu menghapus atribut yang tidak digunakan seperti dalam tabel di bawah ini.

**Tabel 4.2 Penyeleksian atribut data**

<b>Nama atribut</b>	<b>Penting</b>	<b>Keterangan</b>
Id_message	-	Tidak digunakan
Text_message	v	Digunakan
Date_message	-	Tidak digunakan
Id_kategori	-	Tidak digunakan
Account	-	Tidak digunakan
Lattitude	-	Tidak digunakan

Dalam penyeleksian atribut di atas maka didapatkan atribut `text_message` yang akan dipakai dalam proses pengolahan data selanjutnya. Tabel 4.4 adalah contoh data yang akan dipakai pada tahapan selanjutnya :

**Tabel 4.3 Contoh Data Hasil Seleksi Atribut**

<b>text_message</b>
@SapawargaSby min, mintak infonya pengumuman penerimaan karyawan outsourcing dinkominfo kota sby,, matur nuhun,,
@SapawargaSby Setiap jam 12 malam kawasan kutisari indah selalu terpolusi asap, ini sangat mengganggu pernafasan kami. Mohon menyelidikinya
@SapawargaSby min, tlg bantu sebarkan kuesioner ini ke peg.swasta yg kerja di/berasal dari Surabaya ya, matur nuwun.. <a href="http://t.co/guFnF9yL9S">http://t.co/guFnF9yL9S</a>
RT @hentrisaaa: @SapawargaSby min, tlg bantu sebarkan kuesioner ini ke peg.swasta yg kerja di/berasal dari Surabaya ya, matur nuwun.. <a href="http://t.co/guFnF9yL9S">http://t.co/guFnF9yL9S</a>

### 4.3. Pelabelan Kelas Data

Pada tahapan ini akan dilakukan pemberian label kelas pada data mentah yang ada. Label yang diberikan adalah positif, negatif, netral, dan unrelated. Pemberian label positif, negatif, dan netral akan diberikan pada tweet yang mengandung sentimen kata-kata tersebut. Sedangkan untuk label kelas unrelated diberikan pada tweet yang tidak mengandung sentimen manapun dan berisikan tweet yang berada di luar Surabaya atau tidak ada kaitannya. Tabel 4.5 adalah beberapa contoh data yang telah diberi label kelas :

Tabel 4.4 Contoh data yang sudah dilabeli

text_message	klasifikasi
@e100ss slmt sore khususnya buat PDAM kota surabaya knp air PDAM ditmpt sy jl Sidotopo wetan baru gg4 sdh beberapa bln ini g klwr sm sekali?	negatif
@e100ss marmoyo>TL FO wonokromo padat merambat rek	negatif
@e100ss 11:36.kandangan lalin rame lancer ada penertiban PK5.Sdh ada petugas polisi.polsek Benowo yg mengatur <a href="http://t.co/YUXWqZnnFl">http://t.co/YUXWqZnnFl</a>	positif
@e100ss ada kebakaran ilalang di perum kalijudan indah. Pmk sudah ada	positif
@SapawargaSby min, mintakinfonya pengumuman penerimaan karyawan outsourcing dinkominfo kota sby,, matur nuhun,,	netral
r_muhammad_s jambangan macet di perempatan habis spbu kalau tahu arah lain hindari kawan	netral
@e100ss didaerah balongmasin, pungging, mojosari.1 bulan terakhir ada 3 kasus pencurian sapi semua sapi ditemukan mati dan diambil dagingnya	unrelated
RT @ridooetoro: @e100ss purwosari>pandaan>hingga pintu masuk arteri terpantau lancar dua arah	unrelated



#### 4.4. Penghapusan Data

Setelah semua data diberikan label positif, negatif, netral dan unrelated maka langkah selanjutnya adalah menghapus data unrelated yang ada. Data ini dihapus karena tidak ada kaitannya dengan kota Surabaya. Sehingga jumlah data yang akan digunakan dalam tahapan selanjutnya adalah sebanyak 2888 data.

#### 4.5. Praproses Data

Ketika data sudah siap dan terkumpul dalam sebuah basis data, maka selanjutnya dilakukan tahapan praproses data. Hal ini dilakukan untuk mengoptimalkan pembuatan model latih dan model uji klasifikasi. Tahapan praproses yang dilakukan di bawah ini adalah praproses utama yang digunakan pada semua model klasifikasi. Tahap praproses ini dibuat dengan menggunakan tools Rstudio dengan menggunakan library NLP dan library tm.

##### a. Case folding

Pada tahap ini dilakukan dengan menyeragamkan huruf menjadi kecil semua (*lower case*), penghapusan angka dan tanda baca. Contoh data yang telah diproses dengan case folding :

Sebelum dibersihkan	Sesudah dibersihkan
@e100ss Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik :(	@e100ss tdk msk diakal tl perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik :(
<a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4m5rX</a>	<a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4m5rX</a>

## b. Proses cleansing

Pada tahap ini dilakukan dengan membersihkan dokumen dengan menghilangkan karakter HTML, ikon emosi, *username*(@username), dan link(<http://xxx>). Berikut adalah kode program untuk melakukan :

	Sebelum dibersihkan	Sesudah dibersihkan
Karakter HTML	@e100ss menyisir ruas a.yani..rsal<gt;tam anpelangi ramai lancar walau cenderung padat menjelang TL	@e100ss menyisir ruas a.yani..rsal;tamanpel angi ramai lancar walau cenderung padat menjelang TL
Ikon emosi	@e100ss Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik:( <a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4 m5rX</a>	@e100ss Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik <a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4m 5rX</a>
Username	@e100ss Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan	Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik

	lbh,ijonya cm 3detik :( <a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4m5rX</a>	:( <a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4m5rX</a>
Link http	@e100ss Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik :( <a href="http://t.co/aoDZx4m5rX">http://t.co/aoDZx4m5rX</a>	@e100ss Tdk msk diakal TL perempatan papaya,patung kuda sby barat masa merahnya 5menitan lbh,ijonya cm 3detik :(

Tabel 4.5 Data hasil crawling

text message	date message	id_kategori	account	latitude	longitude
@SapawargaSby min, mintak infonya pengumuman penerimaan karyawan outsourcing dinkominfo kota sby,, matur nuhun,,	01/09/2015 20:35	2	@Rizal_M_Vadly	0	0
@SapawargaSby Setiap jam 12 malam kawasan kutisari indah selalu terpolusi asap, ini sangat mengganggu pernafasan kami. Mohon menyelidikinya	02/09/2015 0:17	2	@Susanna83	0	0
@SapawargaSby min, tlg bantu sebar kan kuesioner ini ke peg.swasta yg kerja di/berasal dari Surabaya ya, matur nuwun.. <a href="http://t.co/guFnF9yL9S">http://t.co/guFnF9yL9S</a>	02/09/2015 10:05	2	@hentrisaaa	0	0
RT @hentrisaaa: @SapawargaSby min, tlg bantu sebar kan kuesioner ini ke peg.swasta yg kerja di/berasal dari Surabaya ya, matur nuwun.. <a href="http://t.co/guFnF9yL9S">http://t.co/guFnF9yL9S</a>	02/09/2015 10:37	2	@SapawargaSby	0	0
RT @satpolppsby: Penertiban pkl yang ada d atas pedestrian di frontage road A. Yani @e100ss @SapawargaSby <a href="http://t.co/ISm5GTogKB">http://t.co/ISm5GTogKB</a>	02/09/2015 13:25	2	@SapawargaSby	0	0
@e100ss Tolong ya untuk pegawai bank BRI pabean untuk waktu jam kerja mohon ditepati, skr sdh jam 14.07 wib, jam segini kok masih istirahat	15/09/2015 14:07	2	@edomania2	0	0

## **BAB V**

### **IMPLEMENTASI**

Bab ini menjelaskan mengenai implementasi terkait data yang digunakan. Implementasi yang dilakukan menggunakan metode klasifikasi yaitu Naïve Bayes dan Support Vector Machine (SVM). Pada tahap ini akan meliputi data implementasi, lingkungan implementasi, pelaksanaan dan hasil implementasi, serta hasil dari visualisasi.

#### **5.1. Data Implementasi**

Data yang digunakan dalam penelitian ini baik untuk tahapan pengujian atau pelatihan adalah data yang terkumpul dari hasil crawling 2 akun twitter yaitu @e100ss dan @SapawargaSby [30] selama periode tertentu dalam 1 September 2015 sampai dengan 13 Oktober 2015. Data yang didapatkan yaitu berupa data mentah yang berjumlah sebanyak 5836 data. Dari data mentah ini akan diolah dan hanya akan diambil atribut `text_message` untuk masuk ke proses klasifikasi seperti pada bab 4.2<sup>1</sup>. Atribut yang akan ditambahkan yaitu atribut klasifikasi yang merupakan labelkelas klasifikasi dari dataset yang ada. Data yang digunakan dalam data pelatihan dan data pengujian masing-masing melalui tahapan praproses data yang sama. Untuk pembagian data latih dan data uji dalam penelitian ini yaitu sebesar 70 : 30.

Pada data implementasi yang digunakan dalam proses klasifikasi, terdapat 2 dataset yang akan digunakan pada masing-masing skenario. Dataset 1 adalah dataset awal yang sudah dihapus data *unrelatednya* seperti pada bab 4.4<sup>2</sup> dan bersifat belum baku.

---

<sup>1</sup> Bab 4.2 adalah Bab Perancangan yang berada di halaman 36

<sup>2</sup> Bab 4.4 adalah Bab Perancangan yang berada di halaman 38

Sedangkan dataset 2 adalah dataset yang sudah mengalami proses pembakuan pada masing-masing tweetnya. Proses pembakuan kata-kata pada tweet ini akan dibahas pada bab 5.3.2.5<sup>3</sup>. Berikut adalah distribusi dataset yang digunakan pada penelitian ini :

**Tabel 5.1 Distribusi Penggunaan Dataset**

<b>Dataset</b>	<b>Jumlah dataset</b>	<b>Skenario 1</b>	<b>Skenario 2</b>
Dataset 1	2958	Model 1, 2, dan 3	Model 1 dan 2
Dataset 2	2888	Model 4, 5, 6, dan 7	Model 3, 4, 5, dan 6

## **5.2. Lingkungan Implementasi**

Lingkungan implementasi merupakan kriteria perangkat pengujian yang digunakan untuk menguji sistem yang dibangun pada tugas akhir ini. Lingkungan implementasi terdiri dari perangkat keras dan perangkat lunak. Adapun perangkat keras yang digunakan ditunjukkan pada Tabel 5.1.

**Tabel 5.2 Lingkungan implementasi perangkat keras**

<b>Perangkat keras</b>	<b>Spesifikasi</b>
Jenis	Notebook
Processor	Intel(R) Core(TM) i5-560M CPU@2.66GHz up to 3.20GHz
RAM	4GB

---

<sup>3</sup> Bab 5.3.2.5 adalah Bab Implementasi yang berada di halaman 55

Selain itu, terdapat perangkat lunak yang digunakan dalam pengerjaan tugas akhir ini. Perangkat lunak ini berfungsi sebagai perangkat yang digunakan dalam melakukan pengujian sistem yang dibangun dengan menggunakan 2 algoritma klasifikasi yang ditunjukkan pada Tabel 5.2.

**Tabel 5.2 Lingkungan implementasi perangkat lunak**

<b>Perangkat lunak</b>	<b>Spesifikasi</b>
Sistem Operasi	Windows 10
Bahasa Pemrograman	R
Tools	RStudio

### **5.3. Klasifikasi**

Dalam pengerjaan tugas akhir ini dilakukan proses klasifikasi untuk mengelompokkan data implementasi ke dalam kategori kelas yang telah ditentukan yaitu positif, negatif, dan netral. Untuk mengerjakan klasifikasi ini digunakan 2 algoritma klasifikasi yaitu Naïve Bayes dan Support Vector Machine(SVM). Penggunaan algoritma ini digunakan untuk membandingkan hasil akurasi yang lebih bagus. Nantinya kita dapat melihat mana performa yang lebih baik diantara keduanya. Dalam klasifikasi ini akan dibagi menjadi 2 skenario yaitu pertama pengerjaan klasifikasi dengan menggunakan algoritma Naïve Bayes dan kedua dengan menggunakan algoritma Support Vector Machine(SVM).

#### **5.3.1. Skenario Klasifikasi**

Dalam mengerjakan klasifikasi pada tugas akhir ini dilakukan beberapa skenario. Terdapat 2 skenario yang dilakukan dalam penelitian ini, untuk skenario 1 dikerjakan dengan menggunakan

algoritma Naïve Bayes dan skenario 2 menggunakan algoritma Support Vector Machine (SVM).

Pada skenario 1 terdapat 7 model yang dikerjakan yaitu dengan rincian :

- Model 1 → Dataset 1 + Pengerjaan dilakukan dengan alur algoritma dasar klasifikasi normal tanpa tambahan + Dikerjakan dengan pemberian nilai parameter frequent word
- Model 2 → Dataset 1 + Pengerjaan dilakukan dengan perubahan metode pembobotan kata yaitu dengan menggunakan 3 metode yaitu Tf, Tf-Idf dan Binary
- Model 3 → Dataset 1 + Pengerjaan dilakukan dengan perubahan nilai parameter *frequent words* yang ada di setiap kategori
- Model 4 → Dataset 2 + Pengerjaan dilakukan dengan pemberian 3 metode yaitu Tf, Tf-Idf dan Binary + percobaan nilai parameter *frequent words*
- Model 5 → Dataset 2 + Pengerjaan dilakukan dengan perubahahan pada tahapan praproses data, yaitu akan ditambahkan tahapan yaitu penghapusan kata (stopword removal). Kata-kata yang digunakan diambil dari source data github yang terdapat pada lampiran C.1
- Model 6 → Dataset 2 + Pengerjaan dilakukan dengan menambah stopwords yang digunakan(ditambahkan beberapa stopwords dari kata yang muncul di semua kategori). Stopword yang digunakan terdapat pada lampiran C.2
- Model 7 → Pada model 7 digunakan dengan menggunakan dataset 2, pengerjaan dilakukan dengan perubahahan pada penggunaan stopwords dan penggantian nilai parameter *frequent words*.



1. Dataset 2 + Stopword model 6 + penggantian pembobotan kata(terdapat 9 kata) yang dibobotkan lebih sebagai pengganti nilai parameter *frequent words*
2. Dataset 2 + Stopword model 5 + penggantian pembobotan kata(terdapat 13 kata) yang dibobotkan lebih sebagai pengganti nilai parameter *frequent words*
3. Dataset 2 + Stopword model 6 + penambahan kata-kata stopwords yang digunakan → stopwords model 7
4. Dataset 2 + Stopword model 7 + penggantian pembobotan kata (terdapat 30 kata) yang dibobotkan lebih sebagai pengganti nilai parameter *frequent words*

Pada skenario 2 terdapat 6 model yang dikerjakan yaitu dengan rincian :

- Model 1 → Dataset 1 + Pengerjaan dilakukan dengan alur algoritma dasar klasifikasi normal tanpa tambahan. Untuk nilai C dan  $\gamma$  dalam model masih default yang didapat dari sistem
- Model 2 → Dataset 1 + Pengerjaan dilakukan dengan perubahan metode pembobotan pada model 1, akan dikerjakan dengan 3 metode pembobotan yaitu TF, Tf-Idf, dan Binary
- Model 3 → Dataset 2 + Pengerjaan dilakukan dengan 1 kernel yaitu RBF dengan metode gridsearch (pencarian C dan  $\gamma$  terbaik)
- Model 4 → Dataset 2 + Pengerjaan dilakukan dengan 1 kernel yaitu Linear dengan metode gridsearch (pencarian C dan  $\gamma$  terbaik)
- Model 5 → Dataset 2 + Pengerjaan dilakukan dengan kernel RBF + Metode pembobotan pada model 3, akan dikerjakan dengan 3 metode pembobotan yaitu TF, Tf-Idf, dan Binary. Nilai C yang akan dibandingkan yaitu  $2^{-0.4}$  sampai  $2^{10}$ , dan  $\gamma$  yaitu  $2^{-1.65}$  sampai  $2^{-1}$

- Model 6 → Dataset 2 + Pengerjaan dilakukan dengan kernel Linear + Metode pembobotan pada model 3, akan dikerjakan dengan 3 metode pembobotan yaitu TF, Tf-Idf, dan Binary. Nilai C yang akan dibandingkan yaitu  $2^{-10}$  sampai  $2^{10}$

### 5.3.2. Skenario 1

Pada skenario pertama ini semua model yang dikerjakan akan menggunakan algoritma klasifikasi Naïve Bayes. Di dalam skenario pertama ini terdapat 7 model klasifikasi yang akan dilakukan dengan alur standar yang telah dijelaskan pada bab 5.3.1<sup>4</sup>. Berikut adalah deskripsi dan tujuan pembuatan model klasifikasi :

**Tabel 5.3 Deskripsi dan tujuan masing-masing model**

<b>Skenario 1</b>	<b>Deskripsi model</b>	<b>Tujuan</b>
Model 1	Pengerjaan dilakukan dengan alur algoritma dasar klasifikasi normal tanpa tambahan. Dikerjakan dengan pemberian nilai parameter frequent word	Melihat nilai akurasi standar yang didapatkan jika tanpa perlakuan apapun
Model 2	Pengerjaan dilakukan dengan perubahan metode pembobotan kata yaitu dengan menggunakan 3 metode yaitu Tf, Tf-Idf dan Binary	Melihat perbedaan hasil akurasi jika dilakukan dengan perbedaan metode pembobotan yang digunakan

<sup>4</sup> Bab 5.3.1 adalah Bab Implementasi yang berada di halaman 46

Model 3	Pengerjaan dilakukan dengan perubahan nilai parameter <i>frequent words</i> yang ada di setiap kategori	Karena dengan metode pembobotan berpengaruh terhadap nilai akurasi, maka dalam model ini akan dicoba pemberian nilai parameter <i>frequent words</i> yang berbeda
Model 4	Pengerjaan dilakukan peninjauan ulang tahapan praproses dengan mencoba menyamakan semua kata yang ada dalam setiap data secara manual. Untuk pengerjaan algoritma klasifikasi dikerjakan dengan perpaduan model 2 dan 3 untuk melihat perbandingan hasil akurasi yang didapatkan. Pada model 4 data yang digunakan adalah data yang sudah ditinjau yang akan dipakai pada model selanjutnya	Setelah dilakukan peninjauan, banyak kata yang mempunyai makna yang sama tetapi dengan penulisan yang berbeda sehingga dicoba untuk menyamakan semua kata tersebut apakah akan berpengaruh terhadap nilai akurasi
Model 5	Pengerjaan dilakukan dengan perubahan pada	Penyeragaman kata berpengaruh dan

	<p>tahapan praproses data, yaitu akan ditambahkan tahapan yaitu penghapusan kata (stopword removal). Kata-kata yang digunakan diambil dari source data github.</p>	<p>setelah ditinjau lagi banyak kata yang muncul adalah kata hubung dan konjungsi-konjungsi sehingga dilakukan penghapusan kata-kata tersebut</p>
Model 6	<p>Pengerjaan dilakukan dengan perubahan pada stopwords yang digunakan. Akan ditambahkan beberapa stopwords yang berjenis sama di semua kategori sehingga akan dilakukan proses penghapusan juga</p>	<p>Setelah dilakukan penghapusan kata, menghasilkan nilai akurasi yang lebih bagus sehingga dilakukan percobaan penggantian kata-kata yang dihapus</p>
Model 7	<p>Pengerjaan dilakukan dengan perubahan pada penggunaan stopwords dan penggantian nilai parameter <i>frequent words</i> :</p> <ol style="list-style-type: none"> <li>1. Stopword model 6 + penggantian pembobotan kata (terdapat 9 kata) yang dibobotkan lebih sebagai pengganti nilai</li> </ol>	<p>Penambahan kata-kata yang dihapus tidak menghasilkan peningkatan akurasi yang signifikan, mencoba membobotkan atau menjadikan kamus kata pada kata-kata yang dihapus</p>

	parameter <i>frequent words</i>	
	2. Stopword model 5 + penggantian pembobotan kata(terdapat 13 kata) yang dibobotkan lebih sebagai pengganti nilai parameter <i>frequent words</i>	
	3. Stopword model 6 + penambahan kata-kata stopwords yang digunakan → stopwords model 7	
	4. Stopword model 7 + penggantian pembobotan kata (terdapat 30 kata) yang dibobotkan lebih sebagai pengganti nilai parameter <i>frequent words</i>	

### 5.3.2.1. Alur Pengerjaan Skenario Klasifikasi 1

Dalam Gambar 5.1 akan dibahas alur pengerjaan standar yang akan diterapkan ke dalam masing-masing model klasifikasi dengan menggunakan algoritma Naïve Bayes. Alur ini adalah alur standar

yang akan dilakukan pada 7 model yang telah dibuat. Untuk alur kode program dapat dilihat pada lampiran A1.



**Gambar 5.1 Alur pengerjaan skenario klasifikasi 1**

Penjelasan :

1. Data dimasukkan ke dalam tools R untuk di olah dan berekstensi file .csv
2. Dilakukan agar data dapat diolah ke tahapan selanjutnya
3. Terdapat 4 bentuk utama di dalam praproses data :
  - Tolower
  - removeNumbers
  - removePunctuation
  - stripWhitespace
4. Pembagian proporsi antara data latih dan data uji yaitu sebesar 70 : 30
5. Pengubahan dokumen menjadi suatu representasi yang mudah dikelola lebih lanjut. Pembobotan dilakukan dengan metode Tf(Term Frequency)
6. Pemberian nilai parameter *frequent word* yaitu untuk memberikan bobot lebih pada jumlah kemunculan kata di suatu dokumen. Misalkan pemberian nilai

parameter *frequent word* 50, hal ini berarti bahwa pembobotan akan dilakukan berdasarkan 50 kata teratas yang sering muncul

7. Konversi data pada *Document Term Matrix* ke dalam bentuk kata yes/no
8. Proses klasifikasi dilakukan dengan menggunakan algoritma Naïve Bayes dan menghitung performa klasifikasi dengan cara menghitung nilai akurasi, *precision*, *recall*, dan *f-measure*

#### **5.3.2.2. Model 1**

Pada model pertama adalah tahap uji coba pertama model klasifikasi dimana model akan dikerjakan dengan alur yang normal tanpa fitur tambahan seperti pada Gambar 5.1. Untuk pemberian nilai parameter *frequent word* di awal yaitu sebesar 50. Nilai parameter *frequent word* 50 berarti bahwa kata yang dipakai adalah 50 kata teratas yang sering muncul dalam sebuah dataset.

#### **5.3.2.3. Model 2**

Setelah melihat hasil pada model 1 perlu dilakukan pengujian klasifikasi lagi yaitu dengan membuat model kedua. Model kedua akan dikerjakan dengan alur berbeda terletak pada pemberian metode pembobotan kata. Hal ini dilakukan untuk melihat apakah memang pembobotan Tf sudah cukup optimal atau belum. Perbedaan tahapan ini akan dilakukan pada tahap nomor 5 pada Gambar 5.1.

Metode pembobotan yang dilakukan pada model klasifikasi ini yaitu ada 3 cara. Berikut akan dijelaskan pada Tabel 5.4 tentang fungsi pembobotan yang digunakan.

Tabel 5.4 Jenis Metode Pembobotan

Metode Pembobotan	Definisi
Tf	Melihat frekuensi kemunculan term yang ada pada dokumen.
Tf – Idf	Melihat frekuensi kemunculan sebuah term di dalam sebuah dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut.
Binary	Tidak melihat banyaknya kemunculan term pada sebuah dokumen.

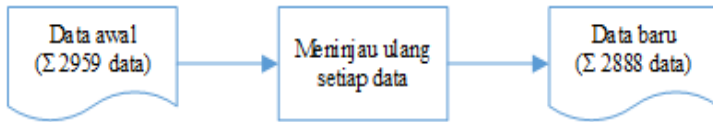
#### 5.3.2.4. Model 3

Setelah melihat pada model kedua, proses klasifikasi yang ada dalam tahap membobotkan kata ternyata dilihat dari penggunaan nilai parameter *frequent words* yang ada di setiap kategori. Sehingga pada model ketiga akan dikerjakan dengan alur yaitu percobaan *frequent words* yang diberikan. Perbedaan tahapan ini akan dilakukan pada tahap nomor 6 pada Gambar 5.1. *Frequent words* yang dicoba yaitu 1 sampai dengan 100 dengan jarak kelipatan 25. Percobaan *frequent words* yang dikerjakan juga akan dibobotkan dengan 3 cara pada model 2.

#### 5.3.2.5. Model 4

Pada model keempat akan dikerjakan dengan alur yaitu meninjau ulang tahapan praproses dengan mencoba menyamakan semua kata yang ada dalam setiap data secara manual.





**Gambar 5.2 Proses Dihilangkan Data Baru**

Peninjauan ulang setiap data dilakukan dengan menghapus tulisan sapawargasby, rt, &amp, dan penyamaan penulisan yang ada. Kata-kata berikut adalah kata yang memiliki frekuensi banyak disingkat dan akan dilakukan proses pembakuan. Untuk melihat kata apa saja yang dilakukan proses pembakuan dapat dilihat pada lampiran E.

1	Tdk	utk	tk	skrg	sdh	bnyk	jl
2	Tlg	sby	thank	mhn	thx	tsb	dpn
3	nopol	lalin	org	tl	klo	hati2	jg
4	jgn	kalo	spd	gmn	spt	dgn	
5	r2	r4	msh	jln	krn	blm	

Proses ini dilakukan untuk melihat apakah hasil yang didapat akan jauh lebih baik karena semua kata yang ada sudah sama semua. Dalam model ini data yang digunakan di awal telah berkurang karena masih ada data unrelated yang ikut dalam proses klasifikasi. Setelah semua kata sama maka langkah selanjutnya dicari hasil akurasi dengan menggunakan perpaduan model 2 dan 3 dengan jarak *frequent words* yang sama.

#### **5.3.2.6. Model 5**

Melihat dari kata-kata yang muncul dan digunakan dalam *frequent words* kurang merepresentasikan masing-masing kategori dan banyak kata hubung yang ada maka pada model kelima akan dikerjakan dengan alur yaitu menambahkan tahapan praproses. Tahapan praproses yang dilakukan adalah penghapusan kata

(stopword removal). Daftar kata yang diambil pada penelitian ini berasal dari github [31] yang terlampir pada lampiran C1.

#### **5.3.2.7. Model 6**

Pada model keenam akan dikerjakan dengan merubah stopwords pada model kelima. Perubahan stopwords dilakukan dengan mendaftar kata-kata yang paling sering muncul dalam setiap kategorinya. Cara mencari kata-kata yang sering muncul dalam setiap kategorinya yaitu dengan menggunakan library(wordcloud). Dalam model 6 ini kata-kata yang dihapus yaitu ditambahkan dengan kata arah, jalan, surabaya, dan macet. Daftar kata yang sering muncul di setiap kategorinya terangkum dalam lampiran C2.

#### **5.3.2.8. Model 7**

Pada model 7 terdapat 4 cara yang dikerjakan, untuk kata-kata yang digunakan di setiap cara dapat dilihat pada lampiran C. Berikut adalah 4 cara yang akan dilakukan pada model 7 :

1. Kata-kata yang sering muncul di setiap kategori akan dibobotkan lebih yaitu menghapus penggunaan *frequent words* di tahapan 6 pada Gambar 5.1. Kata-kata yang akan diberikan bobot lebih yaitu ada 9 kata : lancar, ramai, padat, polisi, tolong, mobil, motor, hari, dan info.
2. Memadukan antara 4 kata yang dihapus pada model 6 yaitu arah, jalan, surabaya, dan macet dengan model 7 yang pertama. Sehingga akan didapatkan 13 kata yang akan dibobotkan lebih.
3. Mendaftar 20 kata teratas dan yang sama akan dicoba dihapus. Untuk 20 kata teratas akan disajikan dalam lampiran. Berikut adalah kata yang ditambahkan pada penggunaan stopwords di model 6.

"lancar", "terimakasih", "arah", "jalan", "light",  
 "surabaya", "polisi", "infonya", "info", "macet", "hari",  
 "tolong", "padat", "untuk"

4. Memberikan dictionary kata bukan dengan nilai-nilai pada *frequent word*. Kata yang digunakan adalah kata yang tersisa dari 20 kata teratas. Kata-kata tersebut adalah

"ramai", "good", "terpantau", "lintas", "traffic", "kasih",  
 "semoga", "dolog", "kebakaran", "air", "wonokromo",  
 "alhamdulillah", "gubsuryo", "raya", "mohon", "mobil",  
 "motor", "kawan", "selamat", "min", "daerah", "pagi",  
 "jam", "pdam", "mati", "lampu", "tolong", "jembatan",  
 "unesa", "pasar"

### 5.3.3. Skenario 2

Pada skenario kedua ini semua model yang dikerjakan akan menggunakan algoritma Support Vector Machine(SVM). Terdapat model klasifikasi yang akan dilakukan dengan alur pengerjaan yang berbeda-beda. Pada algoritma SVM terdapat 4 kernel utama yang bisa digunakan dan pada penelitian ini hanya akan digunakan 2 kernel yaitu kernel RBF dan kernel Linear. Berikut adalah deskripsi dan tujuan pembuatan model klasifikasi pada skenario 2:

**Tabel 5.5 Deskripsi dan ytujuan pembuatan skenario 2**

Skenario 2	Deskripsi model	Tujuan
Model 1	Pengerjaan dilakukan dengan alur algoritma dasar klasifikasi normal tanpa tambahan. Untuk nilai C dan $\gamma$ dalam model masih default yang	Pengerjaan standar, dilakukan untuk membandingkan hasil yang didapat dengan menggunakan

	didapat dari sistem. Data yang digunakan adalah data awal yang digunakan pada skenario 1 model 1	algoritma Naive Bayes
Model 2	Pengerjaan dilakukan dengan perubahan metode pembobotan pada model 1, akan dikerjakan dengan 3 metode pembobotan yaitu TF, Tf-Idf, dan Binary	Melihat perbedaan hasil akurasi jika dilakukan dengan perbedaan metode pembobotan yang digunakan
Model 3	Pengerjaan dilakukan dengan adanya perubahan data yang digunakan, yaitu data sama dengan skenario 1 model 4. Untuk pengerjaan algoritma klasifikasi hanya dilakukan dengan 1 kernel yaitu RBF dengan metode gridsearch (pencarian C dan $\gamma$ ) terbaik	Pencarian nilai akurasi terbaik dengan kernel RBF yang dilakukan dengan cara gridsearch
Model 4	Pengerjaan dilakukan dengan perubahan kernel pada model 3, akan dikerjakan dengan menggunakan kernel Linear. Untuk nilai gridsearch yang	Pencarian nilai akurasi terbaik dengan kernel Linear sebagai pembanding kernel RBF. Pencarian nilai terbaik dilakukan

	digunakan sama dengan model 3	dengan cara gridsearch
Model 5	Pengerjaan dilakukan dengan perubahan metode pembobotan pada model 3, akan dikerjakan dengan 3 metode pembobotan yaitu TF, Tf-Idf, dan Binary. Nilai C yang akan dibandingkan yaitu $2^{-0,4}$ sampai $2^{10}$ , dan $\gamma$ yaitu $2^{-1,65}$ sampai $2^{-1}$	Mencoba mencari hasil yang optimal dengan 3 metode pembobotan, kernel yang digunakan yaitu RBF
Model 6	Pengerjaan dilakukan dengan perubahan metode pembobotan pada model 4, akan dikerjakan dengan 3 metode pembobotan yaitu TF, Tf-Idf, dan Binary. Nilai C yang akan dibandingkan yaitu $2^{-10}$ sampai $2^{10}$	Mencoba mencari hasil yang optimal dengan 3 metode pembobotan, kernel yang digunakan yaitu Linear yang digunakan untuk pembandingan pada model 5

Untuk pengerjaan penelitian ini akan dilakukan dengan menggunakan model GridSearch yaitu mencari perpaduan nilai cost dan gamma untuk mendapatkan hasil klasifikasi terbaik.

Tabel 5.6 Parameter Optimasi SVM

<b>Support Vector Machine (SVM)</b>	<b>Linear</b>	Parameter : Optimasi nilai C
	<b>RBF</b>	Parameter : Optimasi nilai C dan $\gamma$

### 5.3.3.1. Alur Pengerjaan Skenario Klasifikasi 2

Dalam Gambar 5.3 akan dibahas alur pengerjaan standar yang akan diterapkan ke dalam masing-masing model klasifikasi dengan menggunakan algoritma Support Vector Machine(SVM). Untuk lebih jelasnya dapat dilihat pada lampiran.



Gambar 5.3 Alur pengerjaan skenario klasifikasi 2

Penjelasan :

1. Data dimasukkan ke dalam tools R untuk di olah dan berekstensi file .csv
2. Dilakukan agar data dapat diolah ke tahapan selanjutnya dalam bentuk sebuah korpus
3. Terdapat 4 bentuk utama di dalam praproses data :
  - Tolower
  - removeNumbers

- removePunctuation
  - stripWhitespace
4. Pembagian proporsi antara data latih dan data uji yaitu sebesar 70 : 30
  5. Pengubahan dokumen menjadi suatu representasi yang mudah dikelola lebih lanjut. Pembobotan dilakukan dengan metode Tf(Term Frequency)
  6. Konversi data ke dalam *Document Term Matrix*
  7. Nemasukkan model adalah pemilihan jenis kernel (RBF/Linear) dan juga pemasukan nilai parameter C dan  $\gamma$  yang digunakan
  8. Proses klasifikasi dilakukan dengan menggunakan algoritma Support Vector Machine (SVM) dan menghitung performa klasifikasi dengan cara menghitung nilai akurasi, *precision*, *recall*, dan *f-measure*

#### **5.3.3.2. Model 1**

Model 1 dilakukan dengan algoritma dasar klasifikasi dengan menggunakan kernel RBF dan linear tanpa dilakukan konfigurasi tambahan. Model 1 ini dikerjakan dengan perlakuan yang sama pada model 1 skenario 1. Pembobotan yang dilakukan pada model 1 ini yaitu Term Frequency (Tf).

#### **5.3.3.3. Model 2**

Model 2 dilakukan dengan algoritma dasar klasifikasi dengan menggunakan kernel RBF dan linear dengan mengubah metode pembobotannya. Model 2 ini dikerjakan dengan perlakuan yang sama pada model 2 skenario 1. Pembobotan yang dilakukan pada model 1 ini yaitu Tf, Tf-Idf, dan Binary.

#### 5.3.3.4. Model 3

Pada model ketiga, data yang digunakan adalah data yang sudah dibakukan yaitu sejumlah 2888 data dimana akan dikerjakan menggunakan kernel RBF. Di kernel RBF, nilai cost dan gamma mempunyai pengaruh terhadap akurasi sehingga GridSearch yang dilakukan yaitu terlampir pada Tabel 5.8.

**Tabel 5.7 Nilai variabel cost dan gamma model 3**

Variabel	Rentang nilai	Jarak Bilangan
<b>Cost(C)</b>	$2^{-15}$ sampai $2^{-1}$ dan $2^1$ sampai $2^{10}$	1
	$2^{-1}$ sampai $2^{-0,1}$ dan $2^{0,1}$ sampai $2^1$	0,1
	$2^{-0,1}$ sampai $2^0$ dan $2^0$ sampai $2^{0,1}$	0,01
<b>Gamma(<math>\gamma</math>)</b>	$2^{-1,65}$ sampai $2^{1,65}$	0,05

#### 5.3.3.5. Model 4

Pada model kedua akan dikerjakan menggunakan kernel Linear. Di kernel Linear, nilai  $\gamma$  tidak mempunyai pengaruh terhadap akurasi sehingga percobaan GridSearch akan dilakukan pada cost yang terlampir pada Tabel 5.9. Nilai gamma yang akan digunakan yaitu sebesar  $2^{-1,5}$ .

**Tabel 5.8 Nilai variabel cost dan gamma model 4**

Variabel	Rentang nilai	Jarak Bilangan
<b>Cost(C)</b>	$2^{-15}$ sampai $2^{-1}$ dan $2^1$ sampai $2^{15}$	1



Cost(C)	Rentang nilai	Jarak Bilangan
	$2^{-1}$ sampai $2^{-0,1}$ dan $2^{0,1}$ sampai $2^1$	0,1

#### 5.3.3.6. Model 5

Pada model kelima akan dilakukan sama persis dengan model 3. Perbedaan yang terjadi adalah pada metode pembobotan yang digunakan. Jika pada model 3 hanya digunakan metode Tf, maka pada model 5 ini akan dilakukan metode pembobotan dengan menggunakan Tf, Tf-Idf, dan Binary.

#### 5.3.3.7. Model 6

Pada model keenam akan dilakukan sama persis dengan model 4, yang berbeda adalah metode pembobotan yang digunakan. Pada model 4 hanya digunakan metode Tf, sedangkan di model 6 ini akan dilakukan 3 metode pembobotan yaitu dengan menggunakan Tf, Tf-Idf, dan Binary.

### 5.4.Klasifikasi Data Kinerja Pelayanan Publik Kota Surabaya

Dari 2888 data keseluruhan, tweet yang mengandung sentimen terhadap kinerja pelayanan publik di Kota Surabaya yaitu sebanyak 293 data. Pada tabel 5.7 akan dibahas rincian data terkait pelayanan publik di Kota Surabaya.

**Tabel 5.9 Distribusi Pembagian Data Tentang Surabaya**

No	Kelas	Jumlah Data	Data Latih	Data Uji
1	Positif	17	12	5
2	Negatif	153	102	50
3	Netral	124	91	33

Jumlah( $\Sigma$ )	Jumlah Data	Data Latih	Data Uji
	293	205	88

Pada data ini akan dikerjakan dengan 2 metode yaitu dengan Naïve Bayes dan Support Vector Machine (SVM). Model yang dijalankan yaitu :

1. Dengan menggunakan Naïve Bayes
  - Frequent word(1, 5, dan 50), pembobotan 3 metode, dan dengan kombinasi ditambah penggunaan stopwords
  - Frequent word(1, 5, dan 50), pembobotan 3 metode, dan dengan kombinasi tanpa ditambah penggunaan stopwords
2. Dengan menggunakan Support Vector Machine (SVM)
  - Kernel RBF  $C=2^0$  dan  $\gamma=2^{-1,25}$  dengan kombinasi ditambah penggunaan stopwords
  - Kernel RBF  $C=2^0$  dan  $\gamma=2^{-1,25}$  dengan kombinasi tanpa ditambah penggunaan stopwords
  - Kernel Linear  $C=2^0$  dan  $\gamma=2^{-1,25}$  dengan kombinasi ditambah penggunaan stopwords
  - Kernel Linear  $C=2^0$  dan  $\gamma=2^{-1,25}$  dengan kombinasi tanpa ditambah penggunaan stopwords

## 5.5. Visualisasi

Visualisasi merupakan hal yang perlu dilakukan dalam mengolah sebuah data. Untuk mengeksplorasi sebuah data yang besar atau tidak terstruktur maka kita perlu sebuah visualisasi yang berguna memudahkan dalam membaca informasi tersebut dan jauh lebih menarik. Dalam tugas akhir ini akan dilakukan 2 bentuk visualisasi yaitu *wordcloud* dan *streamgraph*. Pembuatan visualisasi ini dikerjakan dengan menggunakan tools R yang nantinya akan dijadikan dalam satu lembar kerja pada aplikasi Shiny.

### **5.5.1. Wordcloud**

Wordcloud adalah salah satu metode untuk menampilkan data teks agar lebih menarik sekaligus mudah dipahami. Dengan menggunakan wordcloud dapat mengetahui frekuensi kata yang sering muncul di ketiga kategori yaitu positif, negatif, dan netral. Kita juga dapat melihat berapa frekuensi kemunculannya misalkan 10 kali muncul, 50 kali muncul, dan sebagainya. Kegunaan dari dibuatnya wordcloud ini yaitu untuk mengetahui kata-kata mana yang sering muncul yang berpengaruh dalam pengerjaan model klasifikasi uji coba. Untuk mengerjakan pada Rshiny dibutuhkan library untuk memanggil fungsi tersebut yaitu `library(wordcloud)` [32] dan `library(RcolorBrewer)` [33].

### **5.5.2. Streamgraph**

Streamgraph adalah salah satu metode dalam visualisasi data yang digunakan dengan melihat lebar, warna, dan waktu dalam sebuah data. Streamgraph merupakan seperti sebuah grafik yang bisa diatur warnanya dan berbentuk 2 dimensi. Streamgraph yang dibuat dalam penelitian ini dapat menampilkan jumlah tweet dari setiap kategori dalam setiap minggunya. Untuk mengerjakan pada Rshiny dibutuhkan library untuk memanggil fungsi tersebut yaitu `library(dplyr)` [34] dan `library(streamgraph)` [35].

### **5.5.3. Shiny**

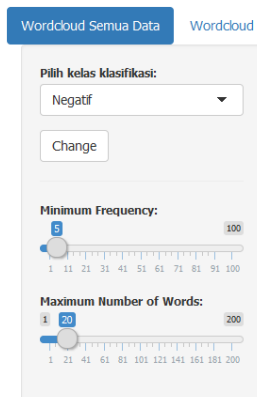
Shiny adalah tools dari Rstudio untuk aplikasi interaktif berbasis web. Shiny bisa digunakan untuk membuat berbagai visualisasi dan simulasi. Tools Shiny yang dilakukan dalam tahapan visualisasi ini adalah untuk menggabungkan 4 bentuk visualisasi yang telah dibuat ke dalam satu lembar kerja. Dalam satu lembar kerja Shiny akan ada tab-tab visualisasi yang telah dikerjakan. Tab pertama yaitu tampilan hasil visualisasi wordcloud semua data,

yang kedua adalah wordcloud data Surabaya, yang ketiga adalah streamgraph semua data, dan yang terakhir adalah streamgraph data Surabaya.

Visualisasi Analisis Sentimen Terhadap Kinerja Pelayanan Publik Di Kota Surabaya Berdasarkan Klasifikasi Komentar Di Media Sosial Dengan Menggunakan Algoritma Naïve Bayes



**Gambar 5.4 Tab Wordcloud**



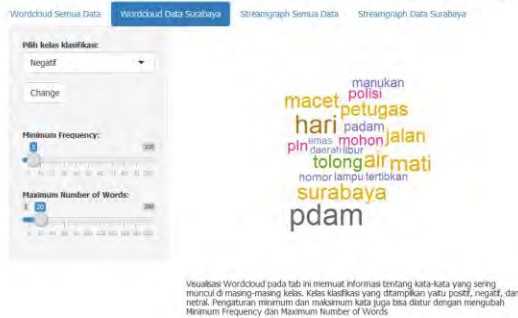
**Gambar 5.5 Menu pilihan kelas klasifikasi semua data**

Pada Gambar 5.4 dan 5.5 di atas merupakan tab pertama yang berisi hasil visualisasi *wordcloud* semua data yang ada dengan jumlah data yaitu 2888. Di sebelah kanan terdapat pilihan untuk

memilih kelas klasifikasi serta minimum dan maksimum frekuensi kata yang ingin ditampilkan dari data. Di sebelah kiri adalah tampilan hasil pemilihan kelas klasifikasi dan frekuensi kata yang dipilih.

Pada Gambar 5.6 dan 5.7 di bawah ini merupakan tab kedua yang berisi hasil *wordcloud* data Surabaya. Tab ini yaitu tab yang akan menampilkan *wordcloud* data tentang yang terkait dengan pelayanan publik di Kota Surabaya. Data yang digunakan yaitu sebanyak 293 data. Kelas klasifikasi yang ditampilkan yaitu sama dengan data yang umum yaitu positif, negatif, dan netral.

Visualisasi Analisis Sentimen Terhadap Kinerja Pelayanan Publik Di Kota Surabaya Berdasarkan Klasifikasi Komentar Di Media Sosial Dengan Menggunakan Algoritma Naive Bayes



**Gambar 5.6 Tab Wordcloud Data Surabaya**

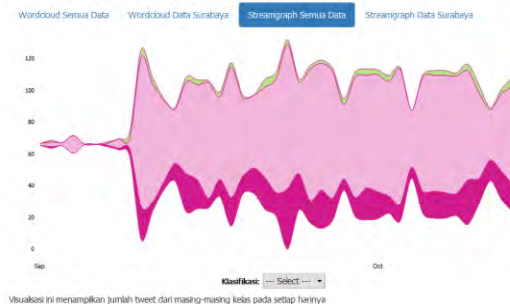
Di sebelah kanan terdapat pilihan untuk memilih kelas klasifikasi serta minimum dan maksimum frekuensi kata yang ingin ditampilkan dari data. Di sebelah kiri adalah tampilan hasil pemilihan kelas klasifikasi dan frekuensi kata yang dipilih.



**Gambar 5.7 Menu Pilihan Kelas Klasifikasi Data Surabaya**

Selanjutnya yaitu pada tab ketiga akan ditampilkan hasil visualisasi *streamgraph* semua data yang ada yaitu berjumlah 2888 data. Pada tab ketiga ini kita hanya memilih kelas klasifikasinya dan grafik hasil *streamgraph* akan tampil di atasnya.

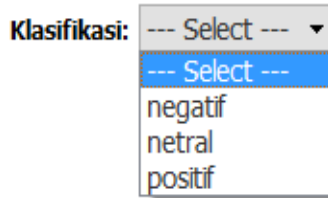
Visualisasi Analisis Sentimen Terhadap Kinerja Pelayanan Publik Di Kota Surabaya Berdasarkan Klasifikasi Komentar Di Media Sosial Dengan Menggunakan Algoritma Naïve Bayes



**Gambar 5.8 Tab Streamgraph Semua Data**

Pada Gambar 5.8 di atas, warna pink tua merupakan hasil dari kelas negatif, sedangkan untuk warna pink muda merupakan hasil dari kelas netral, dan warna hijau merupakan hasil dari kelas positif.

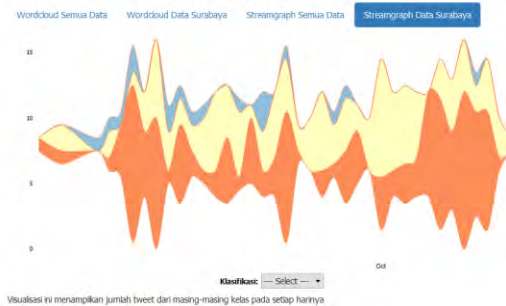
Berikut adalah menu pilihan kelas klasifikasi yang adaa pada tab visualisasi *streamgraph* semua data yang ada.



**Gambar 5.9 Menu Pilihan Kelas Klasifikasi**

Tab yang terakhir yaitu tab yang keempat akan akan ditampilkan hasil visualisasi *streamgraph* data Surabaya yang berjumlah 293 data. Pada tab keempat ini kita hanya memilih kelas klasifikasinya dan grafik hasil *streamgraph* akan tampil di atasnya.

Visualisasi Analisis Sentimen Terhadap Kinerja Pelayanan Publik Di Kota Surabaya Berdasarkan Klasifikasi Komentar Di Media Sosial Dengan Menggunakan Algoritma Naive Bayes



**Gambar 5.10 Tab Streamgraph Data Surabaya**

Pada Gambar 5.10 di atas, warna oranye merupakan hasil dari kelas negatif, sedangkan untuk warna krem atau kuning muda merupakan hasil dari kelas netral, dan warna biru merupakan hasil dari kelas positif. Pada tab ini juga terdapat menu pilihan kelas klasifikasi yang dapat dipilih dan bisa dilihat pada gambar 5.9 di atas.

*Halaman ini sengaja dikosongkan*



## **BAB VI**

### **ANALISA HASIL DAN PEMBAHASAN**

Pada bab ini akan dibahas mengenai hasil dari model klasifikasi yang telah dibangun pada bab implementasi. Hasil klasifikasi ini meliputi perhitungan akurasi, *precision*, dan *recall* dengan pembahasan yang dilakukan mencakup analisis model, performa klasifikasi, dan pembahasan kategori klasifikasi.

#### **6.1. Analisis Model Klasifikasi**

Dalam klasifikasi teks, tujuan yang akan dihasilkan yaitu membuat model yang dapat mengklasifikasi kelas dokumen dengan sesuai dari data yang dipakai. Klasifikasi yang dijalankan pada penelitian ini menggunakan algoritma Naïve Bayes dan Support Vector Machine (SVM). Agar mendapatkan hasil klasifikasi yang baik dilakukan berbagai perlakuan di setiap modelnya. Terdapat 2 skenario yang dijalankan dalam penelitian ini untuk mendapatkan model yang optimal. Proses dalam setiap skenario dapat dilihat pada bab 6.1.1<sup>5</sup> dan 6.1.2<sup>6</sup>.

##### **6.1.1. Hasil Skenario 1**

Tahap implementasi telah dilaksanakan dan menghasilkan berbagai nilai. Skenario 1 ini dijalankan dengan menggunakan algoritma Naïve Bayes. Terdapat 7 model yang dikerjakan dengan alur yang berbeda. Analisis dari hasil implementasi model akan dipaparkan pada berikut ini.

---

<sup>5</sup> Bab 6.1.1 adalah Bab Analisa Hasil dan Pembahasan yang berada di halaman 69

<sup>6</sup> Bab 6.1.2 adalah Bab Analisa Hasil dan Pembahasan yang berada di halaman 76

### 6.1.1.1 Model 1

Model 1 dilakukan dengan algoritma dasar klasifikasi tanpa dilakukan konfigurasi tambahan. Hasil nilai akurasi yang didapat sebesar 76,01%. Dengan nilai akurasi tersebut bukan merupakan model yang optimal, karena hanya dilakukan perlakuan yang standar.

### 6.1.1.2 Model 2

Model 2 dilakukan dengan algoritma yang dikembangkan dari model 1. Algoritma yang dijalankan yaitu mencoba pemberian bobot yang berbeda. Jika pada model 1 dijalankan dengan metode pembobotan Tf, di model 2 ini akan dilakukan pemberian 3 metode pembobotan dengan nilai frequent word yang sama yaitu 50. Hasil menunjukkan bahwa penggunaan metode pembobotan berpengaruh terhadap nilai akurasi.

**Tabel 6.1 Hasil akurasi masing-masing metode pembobotan**

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
50	76,01	77,25	75,67

Didapatkan bahwa pembobotan Tf-Idf yang mempunyai hasil akurasi lebih baik sebanyak 1,24% lebih bagus. Diantara ketiga pembobotan itu hasil binary lah yang memiliki nilai terendah. Pada model 2 hasil yang didapatkan juga belum optimal karena hanya selisih 1,24% sehingga perlu dilakukan percobaan model yang ketiga.

### 6.1.1.3 Model 3

Model 3 dilakukan dengan algoritma yang dikembangkan dari model 2. Algoritma yang dijalankan yaitu mencoba pemberian nilai *frequent word*. Pemberian nilai *frequent word* yang berbeda

akan memberikan pengaruh pada hasil. Hal ini digunakan juga untuk melihat apakah semua metode pembobotan mampu memenuhi semua nilai dan dapat menghasilkan akurasi terbaik. Nilai *frequent words* yang dicoba yaitu 1 sampai dengan 100 dengan jarak kelipatan 25.

**Tabel 6.2 Hasil Akurasi dengan percobaan frequent word**

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
1	58,33333	66,89189	58,33333
5	71,95946	72,63514	71,73423
25	74,43694	76,57658	74,43694
50	76,01351	77,25225	75,67568
75	76,68919	78,04054	77,59009
100	77,36486	-	77,36486

Didapatkan bahwa metode pembobotan Tf-Idf mendapatkan hasil yang terbaik tetapi hanya berhenti sampai nilai *frequent word* 75. Untuk nilai *frequent word* 100, metode Tf-Idf tidak dapat memenuhinya karena tidak adanya jumlah kata yang muncul ketika proses pembobotannya. Berikut adalah hasil jumlah kata yang muncul di setiap frequent word.

**Tabel 6.3 Jumlah kata yang muncul**

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
1	7039	3826	7039
5	864	660	841
25	133	48	126
50	52	11	46

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
75	25	2	21
100	10	0	9

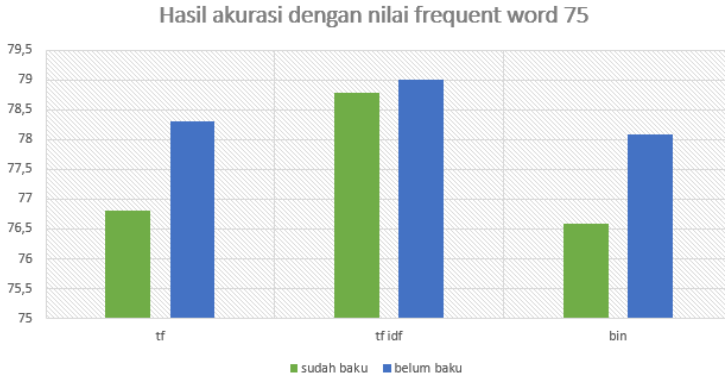
#### 6.1.1.4 Model 4

Pada model keempat akan dikerjakan dengan alur yaitu meninjau ulang tahapan praproses dengan mencoba menyamakan semua kata yang ada dalam setiap data secara manual. Hasil yang didapat akan terangkum dalam Tabel 6.5.

**Tabel 6.4 Hasil Akurasi setelah dibakukan**

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
1	62,51442	69,43483	62,51442
25	73,9331	76,35525	73,9331
50	75,08651	78,66205	74,74048
75	76,81661	78,77739	76,58593
100	78,54671	-	78,54671

Pada Gambar 6.1 dapat dilihat perbedaan hasil akurasi setelah dibakukan dan sebelum dibakukan. Hasil akurasi ini diambil pada nilai *frequent word* 75.



**Gambar 6.1 Perbandingan hasil antara sudah baku dan belum baku**

Untuk mencapai model yang optimal perlu dilakukan analisis lebih lanjut, yaitu mencoba melakukan peninjauan kata apa saja yang muncul pada *frequent word* 100 di metode Tf, Binary, dan pada *frequent word* 75 di metode Tf-Idf.

**Tabel 6.5 Kata yang muncul pada FQ 100**

Metode	Kata-kata
Tf (FQ 100)	ada, apa, arah, bisa, dan dari, depan, hari, ini, jalan, kalau, macet, mohon, padat, polisi, saya, sudah, surabaya, tidak, yang
Binary (FQ 100)	ada, apa, arah, bisa, dan dari, depan, hari, ini, jalan, macet, mohon, padat, polisi, sudah, surabaya, tidak, yang
Tf-Idf (FQ 75)	ada, terimakasih

### 6.1.1.5 Model 5

Melihat kata-kata yang muncul pada model 4 kurang mencerminkan hasil sentimen per kelas klasifikasi dan banyak mengandung kata hubung, maka pada model 5 dilakukan proses penghapusan kata (*stopword removal*). Proses penghapusan kata ini dikerjakan dengan menggunakan aplikasi Rstudio. Untuk daftar kata stopwords bisa dilihat pada lampiran. Pengerjaan algoritma yang dilakukan sama dengan model 4 yaitu percobaan nilai *frequent word* dengan 3 metode pembobotan.

**Tabel 6.6 Hasil Akurasi model 5**

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
1	65,39792	69,20415	65,39792
25	74,5098	76,58593	74,62514
50	76,81661	78,43137	76,81661
75	78,31603	79,00807	78,08535
100	78,89273	-	78,89273

Terjadi perubahan nilai akurasi yang meningkat pada hasil akurasi di ketiga metode pembobotan.

### 6.1.1.6 Model 6

Dari hasil penghapusan stopwords awal + untuk + arah, jalan, surabaya, dan macet maka didapatkan hasil yang cenderung mengalami peningkatan meskipun tidak signifikan pada setiap nilai *frequent word*. Berikut adalah hasil akurasi yang didapatkan dan jumlah kata yang muncul di masing-masing *frequent word* yang telah dijalankan.

Tabel 6.7 Hasil Percobaan model 6

jumlah frekuensi kata	nilai akurasi(persentase)			jumlah kata yang muncul		
	Tf	Tf-IDF	Binary	Tf	Tf-IDF	Binary
<b>1</b>	65,282 56	68,050 75	65,282 58	5496	3924	5496
<b>25</b>	75,317 19	77,970 01	75,432 53	100	57	93
<b>50</b>	77,508 65	78,662 05	77,277 97	36	6	31
<b>75</b>	78,662 05	78,777 39	78,662 05	10	2	8
<b>100</b>	78,662	-	78,662	4	0	4

#### 6.1.1.7 Model 7

Pada model 7 dilakukan berbagai variasi tetapi hasil yang didapat belumlah menunjukkan hasil peningkatan yang signifikan. Hasil yang didapat menunjukkan angka berada di sekitar 78%. Tabel di bawah ini akan menunjukkan masing-masing hasil yang telah dijalankan.

- A. Untuk model dan 2 yaitu memberi kamus kata sendiri sehingga metode pembobotan tidak memiliki pengaruh terhadap akurasi ini. Selain itu hasil yang didapatkan mempunyai nilai yang sama

Tabel 6.8 Model Pengerjaan model 7

Pembobotan lebih pada kata	Nilai akurasi
"lancar", "ramai", "padat", "polisi", "tolong", "mobil", "motor", "hari", "info"	78,77739
"lancar", "ramai", "padat", "polisi", "tolong", "mobil", "motor", "hari", "info", "arah", "jalan", "surabaya", "macet"	78,77739

- B. Setelah mendaftar 20 kata teratas dan dilakukan penghapusan pada kata yang telah dijelaskan pada bab 5.3.2.8<sup>7</sup> didapatkan hasil pada tabel di bawah ini.

Tabel 6.9 Hasil Akurasi dengan model 7

<i>Frequent words</i>	Nilai Akurasi (persentase)		
	Tf	Tf-IDF	Binary
1	62,62976	66,32065	62,62976
25	75,20185	76,93195	75,77855
50	77,85467	78,77739	78,31603
75	78,77739	-	78,77739
100	78,77739	-	78,77739

- C. Setelah diberikan hasil kamus kata sendiri sisa dari kata yang dihapus, hasil yang didapatkan yaitu sama semua untuk semua metode yaitu 78,77739.

<sup>7</sup> Bab 5.3.2.8 adalah Bab Implementasi yang berada di halaman 56



### 6.1.2. Kesimpulan algoritma Naive Bayes

Jika dilihat dari semua model percobaan hingga yang terakhir, hasil akurasi berkisar hanya pada nilai 77 - 78%. Sedangkan dengan model penghapusan dan pembobotan kamus kata sendiri pun tidak ada efek yang signifikan. Hasil akurasi terakhir yang didapatkan yaitu sebesar 78,77739%, sehingga bisa kita simpulkan bahwa nilai akurasi optimal dalam klasifikasi menggunakan algoritma ini yaitu 78,77%. Selanjutnya model ini akan dianalisis performa klasifikasinya pada bab 6.2.1<sup>8</sup>. Secara keseluruhan algoritma Naïve Bayes kurang optimal dalam data yang digunakan dalam penelitian ini. Naive bayes, mengasumsikan bahwa tiap-tiap kata di dalam setiap kategori adalah tidak bergantung satu sama lain.

### 6.1.3. Hasil Skenario 2

Tahap implementasi telah dilaksanakan dan menghasilkan berbagai nilai. Skenario 2 ini dijalankan dengan menggunakan algoritma Support Vector Machine(SVM). Data yang akan digunakan pada percobaan model SVM yaitu terdiri dari 2 data yaitu sebelum dibakukan(2958 data) dan yang sudah dibakukan sejumlah 2888 data. Terdapat 6 model yang dikerjakan dengan alur yang berbeda. Analisis dari hasil implementasi model akan dipaparkan pada berikut ini.

#### 6.1.2.1 Model 1

Hasil yang didapat dengan perlakuan standar yaitu untuk nilai  $C$  dan  $\gamma$  yang dihasilkan yaitu standar tanpa dilakukan perubahan yaitu  $C=1$ ,  $\gamma= 0,0001420656$ . Pada Tabel 6.11 akan disajikan perbedaan hasil akurasi yang dari keduanya.

---

<sup>8</sup> Bab 6.2.1 adalah Bab Analisa Hasil dan Pembahasan yang berada di halaman 84

**Tabel 6.10 Hasil Akurasi awal model 1**

Kernel	Nilai Akurasi
RBF	78,04054
Linear	73,76126

Didapatkan bahwa hasil akurasi yang lebih bagus dikerjakan dengan menggunakan kernel RBF. Selisih yang didapat diantara keduanya yaitu sebesar 4,27%.

### 6.1.2.2 Model 2

Dengan nilai  $C$  dan  $\gamma$  yang standar akan dilakukan proses metode pembobotan *dtm* yaitu dengan 3 metode. Kernel yang dipakai akan digunakan RBF dan Linear. Pada Tabel 6.11 akan disajikan perbedaan hasil akurasi yang dari keduanya.

**Tabel 6.11 Hasil Akurasi dengan metode yang berbeda**

Metode Pembobotan	Nilai Akurasi Pada Kernel	
	RBF	Linear
Tf	78,04054	73,76126
Tf-Idf	78,04054	67,56757
Binary	78,04054	71,95946

Hasil menunjukkan bahwa hasil akurasi yang lebih bagus didapatkan dengan mengerjakan klasifikasi dengan menggunakan kernel RBF. Untuk metode pembobotan yang diberikan hanya berpengaruh pada kernel Linear sedangkan pada kernel RBF sama saja.

### 6.1.2.3 Model 3

Pada model 3, percobaan dilakukan dengan data yang sudah dibakukan yaitu sejumlah 2888 data. Algoritma dasar klasifikasi yang digunakan yaitu dengan menggunakan kernel RBF tanpa dilakukan konfigurasi tambahan. Pada kernel RBF didapatkan

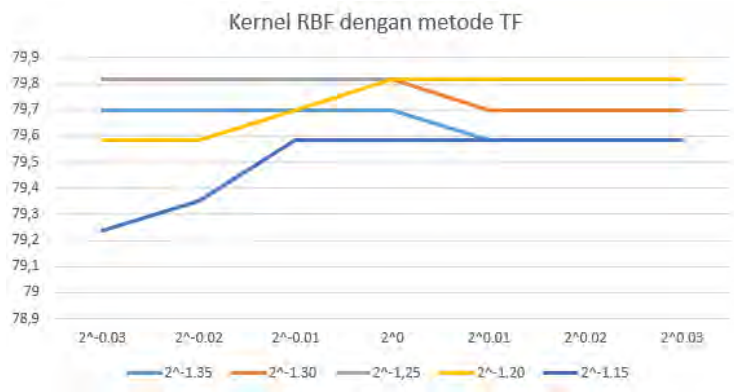
hasil akurasi sebesar 79,81% pada nilai  $C=2^0$  dan  $\gamma=2^{-1.25}$ . Untuk hasil gridsearch yang telah dijelaskan pada Tabel 5.6 akan ditampilkan pada lampiran.

#### **6.1.2.4 Model 4**

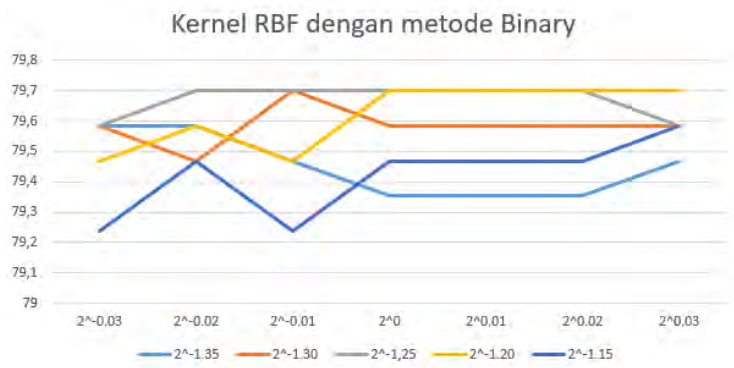
Model 2 dilakukan dengan algoritma dasar klasifikasi dengan menggunakan kernel Linear tanpa dilakukan konfigurasi tambahan. Pada kernel Linear didapatkan hasil akurasi sebesar 71,97% pada nilai  $C=2^0$  dan  $\gamma=2^{-1.25}$ , yang mana nilai  $\gamma$  tidak akan memberikan pengaruh terhadap akurasi. Pada kernel Linear hasil yang didapatkan lebih rendah 7,84%. Hasil gridsearch yang telah dijelaskan pada Tabel 5.7 akan ditampilkan pada lampiran.

#### **6.1.2.5 Model 5**

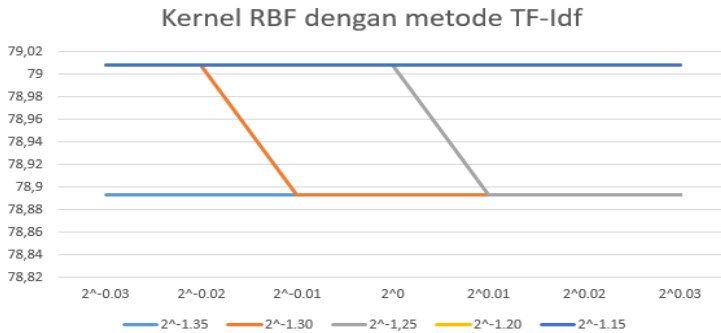
Model 3 dilakukan dengan algoritma dasar klasifikasi dengan menggunakan kernel RBF dengan konfigurasi tambahan yaitu pemberian metode pembobotan. Untuk nilai  $C$  dan  $\gamma$  akan disamakan rentangnya yaitu pada  $C=2^{-0.4}$  sampai  $C=2^{0.4}$  dengan jarak sebanyak 0,1 dan 0,001. Hasil lengkap gridsearch yang telah dijalankan akan ditampilkan pada lampiran. Dengan melihat pada nilai  $C=2^0$  dan  $\gamma=2^{-1.25}$  di dapatkan bahwa hasil metode Tf sebesar 79,81%, metode Tf-Idf sebesar 79%, dan metode Binary sebesar 79,7%. Pada Gambar 6.2, 6.3, dan 6.4 menunjukkan perbedaan hasil akurasi yang didapatkan dengan metode pembobotan yang sama serta nilai  $C$  dan  $\gamma$  yang sama. Pada gambar ini tidak menunjukkan letak akurasi yang berada di tengah melainkan hanya menunjukkan bahwa dengan perpaduan  $C$  dan  $\gamma$  ini hasil dengan Tf memiliki nilai yang lebih besar.



Gambar 6.2 Gridsearch dengan Kernel RBF metode Tf



Gambar 6.3 Gridsearch dengan Kernel RBF metode Binary



**Gambar 6.4** Gridsearch dengan Kernel RBF metode Tf-Idf

### 6.1.2.6 Model 6

Model 6 dilakukan dengan algoritma dasar klasifikasi dengan menggunakan kernel Linear dengan konfigurasi tambahan yaitu pemberian metode pembobotan. Untuk nilai  $C$  dan  $\gamma$  akan disamakan rentangnya yaitu pada  $C=2^{-0.4}$  sampai  $C=2^{0.4}$  dengan jarak sebanyak 0,1 dan 0,001, yang mana nilai  $\gamma$  tidak akan memberikan pengaruh terhadap akurasi. Hasil lengkap gridsearch yang telah dijalankan akan ditampilkan pada lampiran. Dengan melihat pada nilai  $C=2^0$  dan  $\gamma=2^{-1.25}$  di dapatkan bahwa hasil metode Tf sebesar 71,97%, metode Tf-Idf sebesar 60,66%, dan metode Binary sebesar 72,31%.

### 6.1.4. Kesimpulan algoritma SVM

Dengan menggunakan algoritma SVM, hasil yang bagus didapatkan dengan menggunakan kernel RBF tanpa adanya stopwords. Jika dilihat pada nilai  $C$  dan  $\gamma$  yang sama, penggunaan stopwords dan tidak menunjukkan hasil yang sama dan cenderung tidak terlalu berubah. Parameter optimasi *gridsearch* terbaik yang dilakukan terjadi pada range  $C=2^{-0.06}$  sampai  $2^{0.2}$  dan  $\gamma=2^{-1.35}$  sampai  $2^{-1.1}$  dengan hasil akurasi sebesar 79,7 sampai 79,81%.

Hasil dari metode Tf menunjukkan nilai akurasi yang lebih bagus karena setelah dilakukan peninjauan kata terdapat 1 tweet yang memuat kata sama dan diulang yaitu “good good good”. Kata ini mempunyai kelas aktual yaitu netral, jika di Tf kata ini diprediksi sebagai positif. Sedangkan pada metode Tf-Idf dan Binary diprediksi sebagai netral. Hal inilah yang membuat metode Tf mendapatkan akurasi yang bagus.

### 6.1.5 Data Tentang Surabaya

Untuk menilai performa hasil klasifikasi terhadap kinerja pelayanan publik di Kota Surabaya dilakukan implementasi terhadap data yang hanya terkait dengan Kota Surabaya saja.

#### 6.1.5.1. Pengerjaan dengan Naïve Bayes

Pengerjaan pada algoritma Naïve Bayes digunakan dengan percobaan *frequent word* yang dibobotkan dengan 3 metode yaitu Tf, Tf-Idf, dan Binary. Hasil percobaan yang didapatkan yaitu terangkum dalam Tabel 6.12.

**Tabel 6.12 Perbandingan hasil akurasi dengan Naive Bayes**

Pengguna an stopword	Frequent word	Nilai akurasi		
		Tf	Tf-IDF	Binary
Tanpa	1	57,95455	57,95455	57,95455
	5	61,36364	55,68182	60,22727
	50	53,40909	-	53,40909
Dengan	1	60,22727	60,22727	60,22727
	5	52,27273	56,81818	52,27273

	50	56,81818	-	56,81818
--	----	----------	---	----------

Pada metode Tf-Idf tidak bisa menampilkan hasil karena tidak ada kata dalam frequent word yang ikut dalam proseskalsifikasi sehingga tidak menghasilkan nilai dan error.

### 6.1.5.2. Pengerjaan dengan Support Vector Machine

Pengerjaan pada algoritma SVM digunakan dengan penggunaan 2 kernel yaitu RBF dan Linear dan percobaan perlakuan yang sama dengan kernel tersebut yaitu dengan dan tanpa stopwords. Hasil percobaan yang didapatkan yaitu terangkum dalam Tabel 6.13.

**Tabel 6.13 Perbandingan hasil akurasi dengan SVM**

Kernel	Metode pembobotan	Tanpa stopwords	Dengan stopwords
<b>RBF</b>	Tf	59,09091	60,22727
	Tf-Idf	56,81818	56,81818
	Binary	57,95455	59,09091
<b>Linear</b>	Tf	67,04545	62,5
	Tf-Idf	61,36364	61,36364
	Binary	63,63636	63,63636

Pada Tabel 6.13 didapatkan hasil bahwa terjadi perbedaan antara kernel Linear dan RBF. Pada kernel Linear nilai akurasi yang didapatkan lebih tinggi dengan tanpa stopwords sedangkan pada kernel RBF nilai akurasi yang didapatkan lebih tinggi dengan menggunakan stopwords.

## 6.2. Analisis Performa Model Klasifikasi

Analisis performa model klasifikasi dilakukan dengan melihat nilai *precision*, *recall*, dan *f-measure* dari setiap kelas. Pada bagian ini akan dilakukan 2 analisis performa klasifikasi.

### 6.2.1 Skenario 1

Hasil analisis model klasifikasi yang ada pada bab 6.1.1<sup>9</sup>, didapatkan model klasifikasi terbaik yaitu dengan akurasi 78,77%. Dengan akurasi yang optimal belum berarti menunjukkan bahwa memiliki performa yang bagus. Ketika dilihat *confusion matrix*-nya, data tidak dapat terklasifikasi dengan tepat sehingga banyak bermunculan nilai 0, sehingga digunakan model yang lain yaitu dengan akurasi sebesar 78,66%. Nilai akurasi ini didapat dengan metode Tf pada *frequent word* bernilai 75. Berikut adalah tabel *confusion matrix* yang didapatkan.

**Tabel 6.14 *Confusion matrix* terbaik skenario 1**

		Actual		
		negatif	netral	positif
Predicted	Negatif	0	157	1
	Netral	1	682	0
	positif	0	26	0

Setelah di dapatkan hasil *confusion matrix* maka kita coba hitung nilai *precision*, *recall*, dan *f-measure* dari masing-masing kelas dan akan disajikan pada tabel 6.15.

**Tabel 6.15 Nilai *precision*, *recall*, dan *f-measure* skenario 1**

Nilai	Kelas klasifikasi		
	negatif	netral	positif
Precision	0,000000	0,788439	0,000000
Recall	0,000000	0,998536	0,000000
F-Measure	-	0,881137	-

---

<sup>9</sup> Bab 6.1.1 adalah Bab Analisa Hasil dan Pembahasan yang berada di halaman 69



### 6.2.2 Skenario 2

Hasil analisis model klasifikasi yang ada pada bab 6.1.3<sup>10</sup>, mendapatkan model klasifikasi terbaik dengan perpaduan  $C$  dan  $\gamma$  yang terbaik, yaitu pada  $C=2^0$  dan  $\gamma=2^{-1,25}$  sebesar 79,81%. Nilai akurasi tersebut belum tentu yang optimal dan menunjukkan bahwa memiliki performa yang bagus. Tabel 6.16 akan ditampilkan kelas klasifikasi yang tepat dikerjakan dengan melihat pada *confusion matrix*.

**Tabel 6.16 Confusion matrix terbaik skenario 2**

		Actual		
		negatif	netral	positif
Predicted	Negatif	8	0	0
	Netral	150	682	24
	positif	0	1	2

Selanjutnya untuk melihat performa hasil klasifikasi maka dihitung nilai *precision*, *recall*, dan *f-measure*-nya pada Tabel 6.17.

**Tabel 6.17 Nilai precision, recall, dan f-measure skenario 2**

Nilai	Kelas klasifikasi		
	Negative	netral	positif
Precision	0,050633	0,99854	0,076923
Recall	1,000000	0,79673	0,666667

---

<sup>10</sup> Bab 6.1.3 adalah Bab Analisa Hasil dan Pembahasan yang berada di halaman 77

Nilai	Kelas klasifikasi		
	Negative	netral	positif
<b>F-Measure</b>	0,096386	0,886292	0,137931

Berdasarkan hasil yang ditunjukkan pada tabel di atas bahwa kelas netral-lah yang paling mendominasi dalam proses klasifikasi. Ketepatan klasifikasi yang didapat dipengaruhi oleh beberapa faktor, diantaranya jumlah teks atau term yang diidentifikasi, jumlah data latihan yang digunakan, fitur klasifikasi, algoritma yang digunakan, dan kemiripan kata yang ada pada saat proses klasifikasi.

### 6.3. Kinerja Pelayanan Publik di Kota Surabaya

Dalam melihat performa hasil klasifikasi terhadap kinerja pelayanan publik di Kota Surabaya, perlu dilakukan penilaian dengan menggunakan data yang terkait dengan pelayanan publik di Kota Surabaya. Data tersebut mengandung sentimen-sentimen dari per kelasnya yaitu positif, negatif, dan netral. Data yang akan digunakan dalam melihat kinerja pelayanan publik yaitu sebanyak 293 data yang telah dijelaskan pada bab 5 yaitu bab implementasi. Sebanyak 293 data diolah dan dikerjakan dengan skenario 1 dan 2. Konfigurasi hasil skenario terbaik yaitu pada skenario 2 model 2 yaitu penggunaan kernel Linear dalam perhitungannya dan dilakukan tanpa menghapus stopword. Hasil yang didapatkan yaitu tergambar dalam *confusion matrix* berikut.

Tabel 6.18 *Confusion matrix* data Surabaya

		Actual		
		negatif	netral	positif
Predicted	Negatif	33	16	1
	Netral	9	24	0
	positif	1	2	2

Selanjutnya untuk melihat performa hasil klasifikasi maka dihitung nilai *precision*, *recall*, dan *f-measure*-nya.

Tabel 6.19 Nilai *precision*, *recall*, dan *f-measure* data Surabaya

Nilai	Kelas klasifikasi		
	negatif	netral	positif
Precision	0,6600	0,7273	0,40000
Recall	0,7674	0,5714	0,66667
F-measure	0,70966	0,639993	0,500001

Untuk melihat topik apa saja yang dibahas dalam 293 data ini dilakukan dengan cara mencari kata yang sering muncul menggunakan Rstudio library *wordcloud*. Pada tabel 6.20 akan dibahas kata yang sering muncul dalam setiap kategorinya :

Tabel 6.20 Isi dari 15 kata teratas masing-masing kelas

positif		negatif		netral	
word	freq	word	freq	word	freq
Sudah	14	ada	60	yang	46
Ada	13	tidak	55	surabaya	45
polisi	8	yang	49	ada	35
Dan	7	pdam	35	jalan	35
jalan	6	sudah	35	tolong	33

positif		negatif		netral	
word	freq	word	freq	word	freq
kebakaran	4	hari	28	hari	30
mobil	4	air	26	dan	29
roda	4	mati	25	tidak	28
macet	3	surabaya	25	pasar	26
Yang	3	dan	23	mohon	25
Belakang	2	macet	23	sudah	24
Berkat	2	jalan	22	setiap	22
Bisa	2	petugas	22	tanpa	22
Dari	2	ini	21	manukan	21
darmo	2	sampai	20	tama	21

Karena ada kata hubung yang masih terambil sehingga coba dilakukan proses penghapusan pada kata-kata yang sama tersebut agar lebih mencerminkan kategori kelasnya.

**Tabel 6.21** Isi dari 15 kata teratas setelah dilakukan stopwords

positif		negatif		netral	
word	freq	word	freq	word	freq
polisi	8	pdam	35	tolong	33
kebakaran	4	air	26	pasar	26
mobil	4	mati	25	mohon	25
roda	4	petugas	22	manukan	21
belakang	2	tolong	20	tama	21
berkat	2	mohon	15	buka	20
darmo	2	polisi	15	emas	20
dharmahusada	2	manukan	13	libur	20
hijau	2	padam	13	pertokoan	20
kasih	2	nomor	11	tertibkan	20

kiri	2	pln	11	air	19
kpp	2	lampu	10	mati	19
lampu	2	libur	10	pdam	19
lancar	2	tertibkan	10	petugas	13
light	2	baru	9	nomor	10

Dapat disimpulkan bahwa kata-kata yang mewakili masing-masing sentimen yaitu :

**Tabel 6.22 Isi kata dari masing-masing kelas**

Kelas	Kata
<b>Positif</b>	Kebakaran, mobil, roda, belakang, berkat, darmo, dharmahusada, hijau, kasih, kiri, kpp, lampu, lancar, light
<b>Negatif</b>	Padam, pln, lampu, baru
<b>Netral</b>	Pasar, tama, buka, emas, pertokoan

Jika dilihat dari kata-kata yang bermunculan, ada beberapa yang menarik yaitu pada masing-masing kelas :

a. Kelas positif

Kebakaran dan dharmahusada diidentifikasi sebagai kata yang positif karena setelah melihat pada data kata-kata tersebut. Ternyata penggunaan kebakaran dan dharmahusada terletak pada satu kalimat. Sentimen masyarakat cenderung memuji karena kebakaran sudah diatasi dengan adanya mobil pmk yang datang di lokasi. Berikut adalah kalimatnya :

kebakaran di dharmahusada sebuah bengkel dan sudah ada 2 mobil pemadam yang berusaha memadamkan, kebakaran di semacam gudang di jalan dharmahusada terjadi kemacetan sudah ada mobil pmk dan api mulai padam.

b. Kelas negatif

Padam, pln, dan lampu diidentifikasi sebagai kata yang negatif karena setelah melihat pada data kata-kata tersebut. Ternyata kata-kata tersebut diindikasikan bahwa keluhan kepada pln yang sudah lama padam. Untuk tempat-tempat pln yang padam tersebar di daerah karah agung, tenggilis, gayungsari barat, wonocolo, kartini, semampir, bringin, dan tarik. Untuk kata padam yang lainnya adalah lampu yang padam, traffic light yang padam dan ada informasi kebakaran yang mana pmk baru saja meluncur sehingga menimbulkan sentimen negatif dari masyarakat.

c. Kelas netral

Pasar, pertokoan, emas, tama diidentifikasi sebagai kata yang netral karena setelah melihat pada data kata-kata tersebut. Ternyata kata-kata tersebut terdapat dalam satu kalimat yang diindikasikan bahwa informasi kepada pihak terkait untuk menertibkan pertokoan emas yang ada di pasar manukan tama. Selain itu ada juga yang tentang pasar kembang dan rungkut pasar.

## LAMPIRAN A

### KODE PROGRAM

#### A.1 NAIVE BAYES

##### A.1.1 Kode program model 1,2,3

```
bi smi l l ah                                     <-
read.csv("~/cobanai vebayes/awal . csv",          sep=";",
stri ngsAsFactors=FALSE)

library(NLP)
library(tm)

twi tter_corpus                                     <-
Corpus(VectorSource(bi smi l l ah$text_message))
cl ean_corpus      <-      tm_map(twi tter_corpus,
content_transformer(tol ower))
cl ean_corpus      <-      tm_map(twi tter_corpus,
removeNumbers)
cl ean_corpus      <-      tm_map(twi tter_corpus,
removePunctuati on)
cl ean_corpus      <-      tm_map(twi tter_corpus,
stri pWhi tspace)

# tokeni zi ng the corpus
dtm1 <- DocumentTermMatri x(cl ean_corpus) #Tf
dtm1      <-      DocumentTermMatri x(cl ean_corpus,
control =l i st(wei ghti ng=wei ghtTfI df)) #TfI df
dtm1      <-      DocumentTermMatri x(cl ean_corpus,
control =l i st(wei ghti ng=wei ghtBi n)) #bi nary

# di vi de corpus i nto trai ni ng and test data
trai n_si ze <- fl oor(0. 7*nrow(bi smi l l ah))
set. seed(141321)
trai n_i ndi ces <- sampl e(seq_I en(nrow(bi smi l l ah)),
si ze=trai n_si ze)
trai n_corpus  <- cl ean_corpus[ trai n_i ndi ces]
test_corpus    <- cl ean_corpus[ -trai n_i ndi ces]
dtm1_trai n    <- dtm1[trai n_i ndi ces, ]

#I denti fy frequen tly used words
frequen_t_words1 <- fi ndFreqTerms(dtm1_trai n, 1)
frequen_t_words1 <- fi ndFreqTerms(dtm1_trai n, 5)
```

```

frequent_words1 <- findFreqTerms(dtm1_train, 25)
frequent_words1 <- findFreqTerms(dtm1_train, 50)
frequent_words1 <- findFreqTerms(dtm1_train, 75)
frequent_words1 <- findFreqTerms(dtm1_train,
100))
length(frequent_words1)

#Create document-term matrices using frequent
words
twi tter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = frequent_words1))
twi tter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = frequent_words1))

#Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

#Convert document-term matrices
twi tter_train <- apply(twi tter_train, 2,
convert_count)
twi tter_test <- apply(twi tter_test, 2,
convert_count)

#the naive bayes
library(e1071)
twi tter_classifier <- naiveBayes(twi tter_train,
factor(bismiIIah$KLASIFIKASI [train_indices]))

#Evaluate the performance on the test data
predicted <- predict(twi tter_classifier,
twi tter_test)
actual <- bismiIIah$KLASIFIKASI [-train_indices]
100*sum(predicted==actual)/length(predicted)

```

### A.1.2 Code program model 4

```

bismiIIah <-
read.csv("~/cobanai vebayes/bakuhsprt.csv",
sep=";", stringsAsFactors=FALSE)

```



```

library(NLP)
library(tm)
twitter_corpus <-
Corpus(VectorSource(bismitlah$text_message))
clean_corpus <- tm_map(twitter_corpus,
content_transformer(tolower))
clean_corpus <- tm_map(twitter_corpus,
removeNumbers)
clean_corpus <- tm_map(twitter_corpus,
removePunctuation)
clean_corpus <- tm_map(twitter_corpus,
stripWhitespace)

# tokenizing the corpus
dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(weighing=weightTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(weighing=weightBin)) #binary

# divide corpus into training and test data
train_size <- floor(0.7*nrow(bismitlah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismitlah)),
size=train_size)
train_corpus <- clean_corpus[ train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]

#Identify frequently used words
frequent_words1 <- findFreqTerms(dtm1_train, 75)
length(frequent_words1)

#Create document-term matrices using frequent
words
twitter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = frequent_words1))
twitter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = frequent_words1))

#Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

#Convert document-term matrices
twitter_train <- apply(twitter_train, 2,
convert_count)

```

```

twitter_test <- apply(twitter_test, 2,
convert_count)

#the naive bayes
library(e1071)
twitter_classifier <- naiveBayes(twitter_train,
factor(bismi11ah$KLASIFIKASI [train_indices]))

#Evaluate the performance on the test data
predicted <- predict(twitter_classifier,
twitter_test)
actual <- bismi11ah$KLASIFIKASI [-
train_indices]
100*sum(predicted==actual)/length(predicted)

```

### A.1.3 Kode Program model 5

```

bismi11ah <-
read.csv("~/cobanai vebayes/bakuhpsrtnb.csv",
sep=";", stringsAsFactors=FALSE)

library(NLP)
library(tm)

twitter_corpus <-
Corpus(VectorSource(bismi11ah$text_message))
clean_corpus <- tm_map(twitter_corpus,
content_transformer(tolower))
clean_corpus <- tm_map(twitter_corpus,
removeNumbers)
clean_corpus <- tm_map(twitter_corpus,
removePunctuation)
clean_corpus <- tm_map(twitter_corpus,
stripWhitespace)
clean_corpus <- tm_map(twitter_corpus,
removeWords, c(
"ada", "adanya", "adalah", "adapun", "agak",
"agakny", "agar", "akan", "akankah",
"akhirnya", "aku", "akulah", "amat", "amatlah",
"anda", "andalah", "antar", "di antaranya",
"antara", "antaranya", "di antara", "apa",
"apaan", "mengapa", "apabila", "apakah",
"apalagi", "apatah", "atau", "ataukah",
"ataupun", "bagai", "bagai kan", "sebagai",
"sebagainya", "bagaimana", "bagaimanapun",
"sebagai mana", "bagaimanakah", "bagi",
"bahkan", "bahwa", "bahwasanya", "sebaliknya",
"banyak", "sebanyak", "beberapa", "seberapa",
"begini", "begini an", "begini kah", "beginilah",
"sebegini", "begini tu", "begini tukah", "begini tulah",

```

"begi tupun", "sebegi tu", "bel um", "bel uml ah",  
 "sebel um", "sebel umnya", "sebenarnya", "berapa",  
 "berapakah", "berapal ah", "berapapun", "betul kah",  
 "sebetul nya", "bi asa", "bi asanya", "bi la",  
 "bi lakah", "bi sa", "bi sakah", "sebi sanya",  
 "bol eh", "bol ehkah", "bol ehl ah", "buat", "bukan",  
 "bukankah", "bukanl ah", "bukannya", "cuma",  
 "percuma", "dahul u", "dal am", "dan", "dapat",  
 "dari", "dari pada", "dekat", "demi", "demi ki an",  
 "demi ki anl ah", "sedemi ki an", "dengan", "depan",  
 "di", "di a", "di al ah", "di ni", "di ri", "di ri nya",  
 "terdi ri", "dong", "dul u", "ke", "enggak",  
 "enggaknya", "entah", "entahl ah", "terhadap",  
 "terhadapnya",

"hal", "hampi r", "hanya", "hanyal ah", "harus",  
 "harusl ah", "harusnya", "seharusnya", "hendak",  
 "hendakl ah", "hendaknya", "hi ngga", "sehi ngga",  
 "i a", "i al ah", "i barat",

"i ngi n", "i ngi nkah", "i ngi nkan", "i ni", "i ni kah",  
 "i ni l ah", "i tu", "i tukah", "i tul ah", "j angan",  
 "j angankan", "j anganl ah", "j i ka", "j i kal au",  
 "j uga", "j ustru", "kal a", "kal au", "kal aul ah",  
 "kal aupun", "kal i an", "kami", "kami l ah", "kamu",  
 "kamul ah", "kan", "kapan", "kapankah", "kapanpun",  
 "di karenakan", "karena", "karenanya",  
 "kecil", "kemudi an", "kenapa", "kepada",  
 "kepadanya", "keti ka", "seketi ka", "khususnya",  
 "ki ni", "ki ni l ah", "ki ranya", "seki ranya", "ki ta",  
 "ki tal ah", "kok", "l agi", "l agi an", "sel agi",  
 "l ah", "l ai n", "l ai nya", "mel ai nkan", "sel aku",  
 "l al u", "mel al ui", "terl al u", "l ama", "l amanya",  
 "sel ama", "sel amanya", "l ebi h", "terl ebi h",  
 "bermacam", "macam", "semacam", "maka",  
 "makanya", "maki n", "mal ah", "mal ahan", "mampu",  
 "mampukah", "mana", "manakal a", "manal agi",  
 "masi h", "masi hkah", "semasi h", "masi ng", "mau",  
 "maupun", "semaunya", "memang", "mereka",  
 "merekal ah", "meski", "meski pun", "semul a",  
 "munki n", "munki nkah", "nah", "namun", "nanti",  
 "nanti nya", "nyari s", "ol eh", "ol ehnya",  
 "seorang", "seseorang", "pada", "padanya",  
 "padahal", "pal i ng", "sepanj ang", "pantas",  
 "sepantasnya", "sepantasnyal ah", "para", "pasti",  
 "pasti l ah", "pernah", "pul a", "pun", "merupakan",  
 "rupanya", "serupa", "saat", "saatnya", "sesaat",  
 "saj a", "saj al ah", "sal i ng", "bersama", "sama",  
 "sesama", "sambi l", "sampai", "sana", "sangat",  
 "sangatl ah", "saya", "sayal ah", "sebab",  
 "sebabnya", "sebuah", "tersebut", "tersebutl ah",  
 "sedang", "sedangkan", "sedi ki t", "sedi ki tnya",

```
"segal a", "segal anya", "segera", "sesegera",
"sej ak", "sej enak", "sekal i", "sekal i an",
"sekal i pun", "sesekal i", "sekal i gus", "sekarang",
"seki tar", "seki tarnya", "sel a", "sel ai n",

"sel al u", "sel uruh", "sel uruhnya", "semaki n",
"sementara", "sempat", "semua", "semuanya",
"sendi ri", "sendi ri nya", "seol ah", "seperti",
"sepertinya", "seri ng", "seri ngnya", "serta",
"si apa", "si apakah", "si apapun", "di si ni",
"di si ni lah", "si ni", "si ni lah", "sesuatu",
"sesuatunya", "suatu", "sesudah", "sesudahnya", "suda
h", "sudahkah", "sudahl ah", "supaya", "tadi",
"tadi nya", "tak", "tanpa", "setel ah", "tel ah",
"tentang", "tentu", "tentul ah", "tentunya",
"tertentu", "seterusnya", "tapi", "tetapi",
"seti ap", "ti ap", "seti daknya", "ti dak",
"ti dakkah", "ti daki ah", "toh", "waduh", "wah",
"wahai", "sewaktu", "wal au", "wal aupun", "wong",
"yai tu", "yakni", "yang"))
```

#### # tokenizing the corpus

```
dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtBin)) #binary
```

#### # divide corpus into training and test data

```
train_size <- floor(0.7*nrow(bismillah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismillah)),
size=train_size)
train_corpus <- clean_corpus[train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]
```

#### #Identify frequently used words

```
frequent_words1 <- findFreqTerms(dtm1_train, 50)
length(frequent_words1)
```

#### #Create document-term matrices using frequent words

```
twitter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = frequent_words1))
twitter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = frequent_words1))
```

#### #Convert count information to "Yes", "No"

```
convert_count <- function(x) {
y <- ifelse(x > 0, 1, 0)
```

```

y <- factor(y, levels=c(0, 1), labels=c("No",
" Yes"))
}

#Convert document-term matrices
twitter_train <- apply(twitter_train, 2,
convert_count)
twitter_test <- apply(twitter_test, 2,
convert_count)

#the naive bayes
library(e1071)
twitter_classifier <- naiveBayes(twitter_train,
factor(bismillah$KLASIFIKASI[training_indices]))

#Evaluate the performance on the test data
predicted <- predict(twitter_classifier,
twitter_test)
actual <- bismillah$KLASIFIKASI[-training_indices]
100*sum(predicted==actual)/length(predicted)

```

#### A.1.4 Kode program model 6

```

bismillah <-
read.csv("~/cobanai vebayes/bakuhsprtnb.csv",
sep=";", stringsAsFactors=FALSE)

library(NLP)
library(tm)

twitter_corpus <-
Corpus(VectorSource(bismillah$text_message))
clean_corpus <- tm_map(twitter_corpus,
content_transformer(tolower))
clean_corpus <- tm_map(twitter_corpus,
removeNumbers)
clean_corpus <- tm_map(twitter_corpus,
removePunctuation)
clean_corpus <- tm_map(twitter_corpus,
stripWhitespace)
clean_corpus <- tm_map(twitter_corpus,
removeWords, c("ada", "adanya", "adalah",
"adapun", "agak", "agakny", "agar", "akan",
"akankah", "akhirnya", "aku", "akulah", "amat",
"amati", "anda", "andalah", "antar",
"di antaranya", "antara", "antaranya",
"di antara", "apa", "apaan", "mengapa",
"apabila", "apakah", "apalagi", "apatah",
"atau", "ataukah", "ataupun", "bagai",
"bagaimana", "sebagai", "sebagainya",
"bagaimana", "bagaimanapun", "sebagaimana",

```

"bagai manakah", "bagi", "bahkan", "bahwa",  
 "bahwasanya", "sebaliknya", "banyak",  
 "sebanyak", "beberapa", "seberapa", "begi ni",  
 "begi ni an", "begi ni kah", "begi ni lah",  
 "sebegi ni", "begi tu", "begi tukah", "begi tul ah",  
 "begi tupun", "sebegi tu", "bel um", "bel um lah",  
 "sebel um", "sebel umnya", "sebenarnya", "berapa",  
 "berapakah", "berapal ah", "berapanya", "berapapun",  
 "betul kah", "sebetul nya", "bi asa", "bi asanya",  
 "bi la", "bi la kah", "bi sa", "bi sakah", "sebi sanya",  
 "bol eh", "bol ehkah", "bol ehl ah", "buat", "bukan",  
 "bukankah", "bukanl ah", "bukannya", "cuma",  
 "percuma", "dahul u", "dal am", "dan", "dapat",  
 "dari", "dari pada", "dekat", "demi", "demi ki an",  
 "demi ki anl ah", "sedemi ki an", "dengan", "depan",  
 "di", "di a", "di al ah", "di ni", "di ri", "di ri nya",  
 "terdi ri", "dong", "dul u", "ke", "enggak",  
 "enggaknya", "entah", "entahl ah", "terhadap",  
 "terhadapnya", "hal", "hampi r", "hanya",  
 "hanyal ah", "harus", "harusl ah", "harusnya",  
 "seharusnya", "hendak", "hendakl ah", "hendaknya",  
 "hi ngga", "sehi ngga", "i a", "i al ah", "i barat",  
 "i ngi n", "i ngi nkah", "i ngi nkan", "i ni", "i ni kah",  
 "i ni lah", "i tu", "i tukah", "i tul ah", "j angan",  
 "j angankan", "j anganl ah", "j i ka", "j i kal au",  
 "j uga", "j ustru", "kal a", "kal au", "kal aul ah",  
 "kal aupun", "kal i an", "kami", "kami lah", "kamu",  
 "kamul ah", "kan", "kapan", "kapankah", "kapanpun",  
 "di karenakan", "karena", "karenanya",  
 "keci l", "kemudi an", "kenapa", "kepada",  
 "kepadanya", "keti ka", "seketi ka", "khususnya",  
 "ki ni", "ki ni lah", "ki ranya", "seki ranya", "ki ta",  
 "ki tal ah", "kok", "l agi", "l agi an", "sel agi",  
 "l ah", "l ai n", "l ai nnya", "mel ai nkan", "sel aku",  
 "l al u", "mel al ui", "terl al u", "l ama", "l amanya",  
 "sel ama", "sel amanya", "l ebi h", "terl ebi h",  
 "bermacam", "macam", "semacam", "maka",  
 "makanya", "maki n", "mal ah", "mal ahan", "mampu",  
 "mampukah", "mana", "manakal a", "manal agi",  
 "masi h", "masi hkah", "semasi h", "masi ng", "mau",  
 "maupun", "semaunya", "memang", "merek",  
 "merekal ah", "meski", "meski pun", "semul a",  
 "mungki n", "mungki nkah", "nah", "namun", "nant i",  
 "nant i nya", "nyari s", "ol eh", "ol ehnya",  
 "seorang", "seseorang", "pada", "padanya",  
 "padahal", "pal i ng", "sepanj ang", "pantas",  
 "sepantasnya", "sepantasnyal ah", "para", "pasti",  
 "pasti lah", "pernah", "pul a", "pun", "merupakan",  
 "rupanya", "serupa", "saat", "saatnya", "sesaat",  
 "saj a", "saj al ah", "sal i ng", "bersama", "sama",  
 "sesama", "sambi l", "sampai", "sana", "sangat"

```
"sangatlah", "saya", "sayalah", "sebab",
"sebabnya", "sebuah", "tersebut", "tersebutlah",
"sedang", "sedangkan", "sedi ki t", "sedi ki nya",
"segala", "segalanya", "segera", "sesegera",
"sejak", "sejenak", "sekali", "sekali an",
"sekali pun", "sesekali", "sekali gus", "sekarang",
"sekitar", "sekitarnya", "sel a", "sel ai n",
"sel al u", "sel uruh", "sel uruhnya", "semaki n",
"sementara", "sempat", "semua", "semuanya",
"sendiri", "sendirinya", "seolah", "seperti",
"sepertinya", "sering",
```

```
"seringnya", "serta", "si apa", "si apakah",
"si apapun", "di sini", "di sini lah", "si ni",
"si ni lah", "sesuatu", "sesuatunya", "suatu",
"sesudah", "sesudahnya", "sudah",
"sudahkah", "sudahlah", "supaya", "tadi",
"tadinya", "tak", "tanpa", "setelah", "telah",
"tentang", "tentu", "tentulah", "tentunya",
"tertentu", "seterusnya", "tapi", "tetapi",
"setiap", "tiap", "setidaknya", "tidak",
"tidaklah", "tidaklah", "toh", "waduh", "wah",
"wahai", "sewaktu", "walau", "walaupun", "wong",
"untuk", "yaitu", "yakni", "yang", "arah",
"jalan", "surabaya", "macet"))
```

#### # tokenizing the corpus

```
dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(weighing=weightTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(weighing=weightBin)) #binary
```

#### # divide corpus into training and test data

```
train_size <- floor(0.7*nrow(bismlah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismlah)),
size=train_size)
train_corpus <- clean_corpus[train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]
```

#### #Identify frequently used words

```
frequent_words1 <- findFreqTerms(dtm1_train, 50)
length(frequent_words1)
```

#### #Create document-term matrices using frequent words

```
twitter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = frequent_words1))
twitter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = frequent_words1))
```

```

#Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

#Convert document-term matrices
twi tter_train <- apply(twi tter_train, 2,
convert_count)
twi tter_test <- apply(twi tter_test, 2,
convert_count)

#the naive bayes
library(e1071)
twi tter_classifier <- naiveBayes(twi tter_train,
factor(bi smi l l ah$KLASI FI KASI [train_i ndices]))

#Evaluate the performance on the test data
predicted <- predict(twi tter_classifier,
twi tter_test)
actual <- bi smi l l ah$KLASI FI KASI [-train_i ndices]
100*sum(predicted==actual)/length(predicted)

```

### A.1.5 Kode program model 7a

```

bi smi l l ah <-
read.csv("~/cobanai vebayes/baku hpsrtnb.csv",
sep=";", stringsAsFactors=FALSE)

library(NLP)
library(tm)

twi tter_corpus <-
Corpus(VectorSource(bi smi l l ah$text_message))
clean_corpus <- tm_map(twi tter_corpus,
content_transformer(tolower))
clean_corpus <- tm_map(twi tter_corpus,
removeNumbers)
clean_corpus <- tm_map(twi tter_corpus,
removePunctuation)
clean_corpus <- tm_map(twi tter_corpus,
stripWhitespace)
clean_corpus <- tm_map(twi tter_corpus,
removeWords, c("ada", "adanya", "adalah",
"adapun", "agak", "agak nya", "agar", "akan",
"akankah", "akhi r nya", "aku", "aku l ah", "amat",
"amat l ah", "anda", "andal ah", "antar",

```



"di antaranya" , "antara" , "antaranya" ,  
 "di antara" , "apa" , "apaan" , "mengapa" ,  
 "apabila" , "apakah" , "apalagi" , "apatah" ,  
 "atau" , "ataukah" , "ataupun" , "bagai" ,  
 "bagai kan" , "sebagai" , "sebagai nya" ,  
 "bagai mana" , "bagai manapun" , "sebagai mana" ,  
 "bagai manakah" , "bagi" , "bahkan" , "bahwa" ,  
 "bahwasanya" , "sebaliknya" , "banyak" ,  
 "sebanyak" , "beberapa" , "seberapa" , "begi ni" ,  
 "begi ni an" , "begi ni kah" , "begi ni lah" ,  
 "sebegi ni" , "begi tu" ,

"begi tukah" , "begi tul ah" , "begi tupun" ,  
 "sebegi tu" , "bel um" , "bel uml ah" , "sebel um" ,  
 "sebel umnya" , "sebenarnya" , "berapa" ,  
 "berapakah" , "berapal ah" , "berapapun" ,  
 "betul kah" , "sebetul nya" , "bi asa" , "bi asanya" ,  
 "bi la" , "bi lakah" , "bi sa" , "bi sakah" , "sebi sanya" ,  
 "bol eh" , "bol ehkah" , "bol ehl ah" , "buat" , "bukan" ,  
 "bukankah" , "bukanl ah" , "bukannya" , "cuma" ,  
 "percuma" , "dahul u" , "dal am" , "dan" , "dapat" ,  
 "dari" , "dari pada" , "dekat" , "demi" , "demi ki an" ,  
 "demi ki anl ah" , "sedemi ki an" , "dengan" , "depan" ,  
 "di" , "di a" , "di al ah" , "di ni" , "di ri" , "di ri nya" ,  
 "terdi ri" , "dong" , "dul u" , "ke" , "enggak" ,  
 "enggaknya" , "entah" , "entahl ah" , "terhadap" ,  
 "terhadapnya" , "hal" , "hampi r" , "hanya" ,  
 "hanyal ah" , "harus" , "harusl ah" , "harusnya" ,  
 "seharusnya" , "hendak" , "hendakl ah" , "hendaknya" ,  
 "hi ngga" , "sehi ngga" , "i a" , "i al ah" , "i barat" ,  
 "i ngi n" , "i ngi n kah" , "i ngi n kan" , "i ni" , "i ni kah" ,  
 "i ni lah" , "i tu" , "i tukah" , "i tul ah" , "jangan" ,  
 "j angankan" , "j anganl ah" , "j i ka" , "j i kal au" ,  
 "j uga" , "j ustru" , "kal a" , "kal au" , "kal aul ah" ,  
 "kal aupun" , "kal i an" , "kami" , "kami lah" , "kamu" ,  
 "kamul ah" , "kan" , "kapan" , "kapankah" , "kapanpun" ,  
 "di karenakan" , "karena" , "karenanya" , "keci l" ,  
 "kemudi an" , "kenapa" , "kepada" , "kepadanya" ,  
 "keti ka" , "seketi ka" , "khususnya" , "ki ni" ,  
 "ki ni lah" , "ki ranya" , "seki ranya" , "ki ta" ,  
 "ki tal ah" , "kok" , "l agi" , "l agi an" , "sel agi" ,  
 "l ah" , "l ai n" , "l ai nnya" , "mel ai nkan" , "sel aku" ,  
 "l al u" , "mel al ui" , "terl al u" , "l ama" , "l amanya" ,  
 "sel ama" , "sel amanya" , "l ebi h" , "terl ebi h" ,  
 "bermacam" , "macam" , "semacam" , "maka" ,  
 "makanya" , "maki n" , "mal ah" , "mal ahan" , "mampu" ,  
 "mampukah" , "mana" , "manakal a" , "manal agi" ,  
 "masi h" , "masi hkah" , "semasi h" , "masi ng" , "mau" ,  
 "maupun" , "semaunya" , "memang" , "mereka" ,  
 "merekal ah" , "meski" , "meski pun" , "semul a" ,  
 "mungki n" , "mungki n kah" , "nah" , "namun" , "nanti" ,  
 "nanti nya" , "nyari s" , "ol eh" , "ol ehnya" ,

```
"seorang", "seseorang", "pada", "padanya",
"padahal", "pal i ng", "sepanj ang", "pantas",
"sepantasnya", "sepantasnyal ah", "para", "pasti",
"pastilah", "pernah", "pula", "pun", "merupakan",
"rupanya", "serupa", "saat", "saatnya", "sesaat",
"saja", "sajal ah", "sal i ng", "bersama", "sama",
"sesama", "sambi l", "sampai", "sana", "sangat",
"sangatlah", "saya", "sayal ah", "sebab",
"sebabnya", "sebuah", "tersebut", "tersebutl ah",
"sedang", "sedangkan", "sedi ki t", "sedi ki tnya",
"segal a", "segal anyal ah", "segera", "sesegera",
"sej ak", "sej enak", "sekal i", "sekal i an",
"sekal i pun", "sesekal i", "sekal i gus", "sekarang",
"seki tar", "seki tarnya", "sel a", "sel ai n",
"sel al u", "sel uruh", "sel uruhnya", "semaki n",
"sementara", "sempat", "semua", "semuanya",
"sendi ri", "sendi ri nya", "seol ah", "seperti",
"seperti nya", "seri ng", "seri ngnya", "serta",
"si apa", "si apakah", "si apapun", "di si ni",
"di si ni lah", "si ni", "si ni lah", "sesuatu",
"sesuatunya", "suatu", "sesudah", "sesudahnya",
"sudah", "sudahkah", "sudahl ah", "supaya", "tadi",
"tadi nya", "tak", "tanpa", "setel ah", "tel ah",
"tentang", "tentu", "tentul ah", "tentunya",
"tertentu", "seterusnya", "tapi", "tetapi",
"seti ap", "ti ap", "seti daknya", "ti dak",
"ti dakkah", "ti daki lah", "toh", "waduh", "wah",
"wahai", "sewaktu", "wal au", "wal aupun", "wong",
"untuk", "yai tu", "yakni", "yang", "arah",
"jal an", "surabaya", "macet"))
```

#### # tokenizing the corpus

```
dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtBin)) #binary
```

#### # divide corpus into training and test data

```
train_size <- floor(0.7*nrow(bismitlah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismitlah)),
size=train_size)
train_corpus <- clean_corpus[ train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]
```

#### #Create document-term matrices using frequent words

```
twitter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = "lancar", "ramai",
```

```

"padat", "polisi", "tolong", "mobil", "motor",
"hari", "info"))
twitter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = "lancar", "ramai",
"padat", "polisi", "tolong", "mobil", "motor",
"hari", "info"))

#Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

#Convert document-term matrices
twitter_train <- apply(twitter_train, 2,
convert_count)
twitter_test <- apply(twitter_test, 2,
convert_count)

#the naive bayes
library(e1071)
twitter_classifier <- naiveBayes(twitter_train,
factor(bismillah$KLASIFIKASI[train_indices]))

#Evaluate the performance on the test data
predicted <- predict(twitter_classifier,
twitter_test)
actual <- bismillah$KLASIFIKASI[-train_indices]
100*sum(predicted==actual)/length(predicted)

```

### A.1.6 Kode program model 7b

```

bismillah <-
read.csv("~/cobanavebayes/bakuhsprtnb.csv",
sep=";", stringsAsFactors=FALSE)

library(NLP)
library(tm)

twitter_corpus <-
Corpus(VectorSource(bismillah$text_message))
clean_corpus <- tm_map(twitter_corpus,
content_transformer(tolower))
clean_corpus <- tm_map(twitter_corpus,
removeNumbers)
clean_corpus <- tm_map(twitter_corpus,
removePunctuation)
clean_corpus <- tm_map(twitter_corpus,
stripWhitespace)

```

```

clean_corpus <- tm_map(twitter_corpus,
removeWords, c("ada", "adanya", "adalah",
"adapun", "agak", "agakny", "agar", "akan",
"akankah", "akhirnya", "aku", "akulah", "amat",
"amatlah", "anda", "andalah", "antar",
"di antaranya", "antara", "antaranya",
"di antara", "apa", "apaan", "mengapa",
"apabila", "apakah", "apalagi", "apatah",
"atau", "ataukah", "ataupun", "bagai",
"bagai kan", "sebagai", "sebagai nya",
"bagai mana", "bagai manapun", "sebagai mana",
"bagai manakah", "bagi", "bahkan", "bahwa",
"bahwasanya", "sebaliknya", "banyak",
"sebanyak", "beberapa", "seberapa", "begini",
"begini an", "begini kah", "begini lah",
"sebegini", "sebegini tu", "sebegini tul ah",
"begini tupun", "sebegini tu", "belum", "umlah",
"sebelum", "sebelumnya", "sebenarnya", "berapa",
"berapakah", "berapalah", "berapapun",
"betul kah", "sebetulnya", "biasa", "biasanya",
"bi la", "bi la kah", "bi sa", "bi sa kah", "sebi sanya",
"bol eh", "bol eh kah", "bol eh lah", "buat", "bukan",
"bukankah", "bukan lah", "bukannya", "cuma",
"percuma", "dahulu", "dalam", "dan", "dapat",
"dari", "dari pada", "dekat", "demi", "demi ki an",
"demi ki an lah", "sedemi ki an", "dengan", "depan",
"di", "di a", "di al ah", "di ni", "di ri", "di ri nya",
"terdiri", "dong", "dul u", "ke", "enggak",
"enggaknya", "entah", "entah lah", "terhadap",
"terhadapnya", "hal", "hampir", "hanya",
"hanyalah", "harus", "harus lah", "harusnya",
"seharusnya", "hendak", "hendak lah", "hendaknya",
"hi ngga", "sehi ngga", "ia", "ial ah", "i barat",
"i ngi n", "i ngi n kah", "i ngi n kan", "ini", "ini kah",
"ini lah", "itu", "i tukah", "i tul ah", "jangan",
"j angankan", "j angan lah", "jika", "jika lah",
"juga", "justru", "kal a", "kal au", "kal aul ah",
"kal aupun", "kal i an", "kami", "kami lah", "kamu",
"kamulah", "kan", "kapan", "kapankah", "kapanpun",
"di karenakan", "karena", "karenanya", "kecil",
"kemudi an", "kenapa", "kepada", "kepadanya",
"keti ka", "seketi ka", "khususnya", "kini",
"kini lah", "ki ranya", "seki ranya", "ki ta",
"ki tal ah", "kok", "lagi", "lagi an", "sel agi",
"lah", "lain", "lain nya", "mel ai nkan", "sel aku",
"lal u", "mel al ui", "terl al u", "lama", "lamanya",
"sel ama", "sel amanya", "lebi h", "terlebi h",
"bermacam", "macam", "semacam", "maka",
"makanya", "makin", "mal ah", "mal ahan", "mampu",
"mampukah", "mana", "manakal a", "manalagi",
"masih", "masih kah", "semasih", "masih ng", "mau"

```

"maupun", "semaunya", "memang", "mereka",  
 "merekal ah", "meski", "meski pun", "semul a",  
 "mungkin", "mungkin ah", "nah", "namun", "nanti",  
 "nantinya", "nyaris", "ol eh", "ol ehnya",  
 "seorang", "seseorang", "pada", "padanya",  
 "padahal", "pal ing", "sepanjang", "pantas",  
 "sepantasnya", "sepantasnyal ah", "para", "pasti",  
 "pastilah", "pernah", "pula", "pun", "merupakan",  
 "rupanya", "serupa", "saat", "saatnya", "sesaat",  
 "saja", "saj al ah", "sal ing", "bersama", "sama",  
 "sesama", "sambi l", "sampai", "sana", "sangat",  
 "sangatl ah", "saya", "sayal ah", "sebab",  
 "sebabnya", "sebuah", "tersebut", "tersebutl ah",  
 "sedang", "sedangkan", "sedi ki t", "sedi ki tnya",  
 "segal a", "segal anya", "segera", "sesegera",  
 "sejak", "sej enak", "sekal i", "sekal i an",  
 "sekal i pun", "sesekal i", "sekal i gus", "sekarang",  
 "seki tar", "seki tarnya", "sel a", "sel ai n",  
 "sel al u", "sel uruh", "sel uruhnya", "semaki n",  
 "sementara", "sempat", "semua", "semuanya",  
 "sendi ri", "sendi ri nya", "seol ah", "seperti",  
 "seperti nya", "seri ng", "seri ngnya", "serta",  
 "si apa", "si apakah", "si apapun", "di si ni",  
 "di si ni l ah", "si ni", "si ni l ah", "sesuatu",  
 "sesuatunya", "suatu", "sesudah", "sesudahnya",  
 "sudah", "sudahkah", "sudahl ah", "supaya", "tadi",  
 "tadi nya", "tak", "tanpa", "setel ah", "tel ah",  
 "tentang", "tentu", "tentul ah", "tentunya",  
 "tententu", "seterusnya", "tapi", "tetapi",  
 "seti ap", "ti ap", "seti daknya", "ti dak",  
 "ti dakkah", "ti dakl ah", "toh", "waduh", "wah",  
 "wahai", "sewaktu", "wal au", "wal aupun", "wong",  
 "untuk", "yai tu", "yakni", "yang"))

#### # tokenizing the corpus

```
dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(weighing=weightTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(weighing=weightBin)) #binary
```

#### # divide corpus into training and test data

```
train_size <- floor(0.7*nrow(bismlah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismlah)),
size=train_size)
train_corpus <- clean_corpus[train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]
```

#### #Create document-term matrices using frequent words

```

twitter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = "lancar", "ramai",
"padat", "polisi", "tolong", "mobil", "motor",
"hari", "info", "arah", "jalan", "surabaya",
"macet"))
twitter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = "lancar", "ramai",
"padat", "polisi", "tolong", "mobil", "motor",
"hari", "info", "arah", "jalan", "surabaya",
"macet"))

# Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

# Convert document-term matrices
twitter_train <- apply(twitter_train, 2,
convert_count)
twitter_test <- apply(twitter_test, 2,
convert_count)

#the naive bayes
library(e1071)
twitter_classifier <- naiveBayes(twitter_train,
factor(bismiilah$KLASIFIKASI[train_indices]))

#Evaluate the performance on the test data
predicted <- predict(twitter_classifier,
twitter_test)
actual <- bismiilah$KLASIFIKASI[-train_indices]
100*sum(predicted==actual)/length(predicted)

```

### A.1.7 Code program model 7c

```

bismiilah <-
read.csv("~/cobanai vebayes/bakuhpsrtnb.csv",
sep=";", stringsAsFactors=FALSE)

library(NLP)
library(tm)

twitter_corpus <-
Corpus(VectorSource(bismiilah$text_message))
clean_corpus <- tm_map(twitter_corpus,
content_transformer(tolower))

```

```

clean_corpus <- tm_map(twitter_corpus,
removeNumbers)
clean_corpus <- tm_map(twitter_corpus,
removePunctuation)
clean_corpus <- tm_map(twitter_corpus,
stripWhitespace)
clean_corpus <- tm_map(twitter_corpus,
removeWords, c("ada", "adanya", "adalah",
"adapun", "agak", "agaknya", "agar", "akan",
"akankah", "akhirnya", "aku", "akulah", "amat",
"amatlah", "anda", "andalah", "antar",
"di antaranya", "antara", "antaranya",
"di antara", "apa", "apaan", "mengapa",
"apabila", "apakah", "apalagi", "apatah",
"atau", "ataukah", "ataupun", "bagai",
"bagai kan", "sebagai", "sebagai nya",
"bagaimana", "bagaimanapun", "sebagaimana",
"bagaimanakah", "bagi", "bahkan", "bahwa",
"bahwasanya", "sebaliknya", "banyak",
"sebanyak", "beberapa", "seberapa", "begi ni",
"begi ni an", "begi ni kah", "begi ni lah",
"sebegi ni", "begitu", "begitukah", "begitulah",
"begitupun", "sebegitu", "belum", "belumlah",
"sebelum", "sebelumnya", "sebenarnya", "berapa",
"berapakah", "berapalah", "berapapun",
"betulkah", "sebetulnya", "biasa", "biasanya",
"bila", "bilakah", "bisakah", "sebisanya",
"bolah", "bolekah", "bolehlah", "buat", "bukan",
"bukankah", "bukannya", "cuma",
"percuma", "dahulu", "dalam", "dan", "dapat",
"dari", "dari pada", "dekat", "demi", "demi kian",
"demi kianlah", "sedemi kian", "dengan", "depan",
"di", "di a", "di al ah", "di ni", "di ri", "di ri nya",
"terdiri", "dong", "dulunya", "ke", "enggak",
"enggaknya", "entah", "entahlah", "terhadap",
"terhadapnya", "hal", "hampir", "hanya",
"hanyalah", "harus", "haruslah", "harusnya",
"seharusnya", "hendak", "hendaklah", "hendaknya",
"hi ngga", "sehi ngga", "ia", "ialah", "ibarat",
"ingin", "inginkah", "inginkan", "ini", "inikah",
"ini lah", "itu", "itukah", "itulah", "jangan",
"jangankan", "jangankah", "jika", "jikalau",
"juga", "justru", "kala", "kalau", "kalaulah",
"kalapun", "kali an", "kami", "kami lah", "kamu",
"kamulah", "kan", "kapan", "kapankah", "kapanpun",
"di karenakan", "karena", "karenanya", "kecil",
"kemudian", "kenapa", "kepada", "kepadanya",
"ketika", "seketika", "khususnya", "kini",
"kinilah", "kiranya", "sekiranya", "kita",
"kitalah", "kok", "lagi", "lagi an", "selagi",
"lah", "lain", "lainnya", "melainkan", "selaku",
"lalu", "melalui", "terlalu", "lama", "lamanya",

```

"sel ama", "sel amanya", "l ebi h", "terl ebi h",  
 "bermacam", "macam", "semacam", "maka",  
 "makanya", "maki n", "mal ah", "mal ahan", "mampu",  
 "mampukah", "mana", "manakal a", "manal agi",  
 "masi h", "masi hkah", "semasi h", "masi ng", "mau",  
 "maupun", "semaunya", "memang", "mereka",  
 "merekal ah", "meski", "meski pun", "semul a",  
 "mungki n", "mungki nkah", "nah", "namun", "nanti",  
 "nanti nya", "nyari s", "ol eh", "ol ehnya",  
 "seorang", "seseorang", "pada", "padanya",  
 "padahal", "pal i ng", "sepanj ang", "pantas",  
 "sepantasnya", "sepantasnyal ah", "para", "pasti",  
 "pasti lah", "pernah", "pul a", "pun",  
 "merupakan", "rupanya", "serupa", "saat",  
 "saatnya", "sesaat", "saj a", "saj al ah", "sal i ng",  
 "bersama", "sama", "sesama", "sambi l", "sampai",  
 "sana", "sangat", "sangatl ah", "saya", "sayal ah",  
 "sebab", "sebabnya", "sebuah", "tersebut",  
 "tersebutl ah", "sedang", "sedangkan", "sedi ki t",  
 "sedi ki tnya", "segal a", "segal anya", "segera",  
 "sesegera", "sej ak", "sej enak", "sekal i",  
 "sekal i an", "sekal i pun", "sesekal i", "sekal i gus",  
 "sekarang", "seki tar", "seki tarnya", "sel a",  
 "sel ai n", "sel al u", "sel uruh", "sel uruhnya",  
 "semaki n", "sementara", "sempat", "semua",  
 "semuanya", "sendi ri", "sendi ri nya", "seol ah",  
 "seperti", "seperti nya", "seri ng", "seri ngnya",  
 "serta", "si apa", "si apakah", "si apapun",  
 "di si ni", "di si ni l ah", "si ni", "si ni l ah",  
 "sesuatu", "sesuatunya", "suatu", "sesudah",  
 "sesudahnya", "sudah", "sudahkah", "sudahl ah",  
 "supaya", "tadi", "tadi nya", "tak", "tanpa",  
 "setel ah", "tel ah", "tentang", "tentu",  
 "tentul ah", "tentunya", "tertentu",  
 "seterusnya", "tapi", "tetapi", "seti ap", "ti ap",  
 "seti daknya", "ti dak", "ti dakkah", "ti daki ah",  
 "toh", "wadah", "wah", "wahai", "sewaktu",  
 "wal au", "wal aupun", "wong", "untuk", "yai tu",  
 "yakni", "yang", "untuk", "lancar", "teri makasi h",  
 "arah", "jal an", "l i ght", "surabaya", "pol i si",  
 "i nfonya", "i nfo", "macet", "hari", "tol ong",  
 "padat"))

#### # tokenizing the corpus

```
dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtBin)) #binary
```

#### # divide corpus into training and test data



```

train_size <- floor(0.7*nrow(bismitah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismitah)),
size=train_size)
train_corpus <- clean_corpus[ train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]

#Identify frequently used words
frequent_words1 <- findFreqTerms(dtm1_train, 1)
frequent_words1 <- findFreqTerms(dtm1_train, 25)
frequent_words1 <- findFreqTerms(dtm1_train, 50)
frequent_words1 <- findFreqTerms(dtm1_train, 75)
frequent_words1 <- findFreqTerms(dtm1_train, 100)
length(frequent_words1)
#Create document-term matrices using frequent
words
twitter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = frequent_words1))
twitter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = frequent_words1))

# Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

# Convert document-term matrices
twitter_train <- apply(twitter_train, 2,
convert_count)
twitter_test <- apply(twitter_test, 2,
convert_count)

# the naive bayes
library(e1071)
twitter_classifier <- naiveBayes(twitter_train,
factor(bismitah$KLASIFIKASI [train_indices]))

#Evaluate the performance on the test data
predicted <- predict(twitter_classifier,
twitter_test)
actual <- bismitah$KLASIFIKASI [-train_indices]
100*sum(predicted==actual)/length(predicted)

```

### A.1.8 Kode program model 7d

```

bismitah <-
read.csv("~/cobanavebayes/bakuhpsrtnb.csv",
sep=";", stringsAsFactors=FALSE)

```

```

library(NLP)
library(tm)

twitter_corpus <- Corpus(VectorSource(bismillah$text_message))
clean_corpus <- tm_map(twitter_corpus,
content_transformer(tolower))
clean_corpus <- tm_map(twitter_corpus,
removeNumbers)
clean_corpus <- tm_map(twitter_corpus,
removePunctuation)
clean_corpus <- tm_map(twitter_corpus,
stripWhitespace)
clean_corpus <- tm_map(twitter_corpus,
removeWords, c(
"ada", "adanya", "adalah", "adapun", "agak",
"agakny", "agar", "akan", "akankah",
"akhirnya", "aku", "akulah", "amat", "amatlah",
"anda", "andalah", "antar", "di antaranya",
"antara", "antaranya", "di antara", "apa",
"apaan", "mengapa", "apabila", "apakah",
"apalagi", "apatah", "atau", "ataukah",
"alaupun", "bagai", "bagikan", "sebagai",
"sebagai nya", "bagaimana", "bagaimana pun",
"sebagai mana", "bagaimana kah", "bagi",
"bahkan", "bahwa", "bahwasanya", "sebaliknya",
"banyak", "sebanyak", "beberapa", "seberapa",
"begini", "begini an", "begini kah", "beginilah",
"sebegini", "begini tu", "begini tukah", "begini tu lah",
"beginiupun", "sebegini tu", "belum", "belum lah",
"sebelum", "sebelumnya", "sebenarnya", "berapa",
"berapakah", "berapalah", "berapapun",
"betulkah", "sebetulnya", "biasa", "biasanya",
"bila", "bilakah", "bi sa", "bisakah", "sebisanya",
"bolah", "bolakah", "bolehlah", "buat", "bukan",
"bukankah", "bukanlah", "bukannya", "cuma",
"percuma", "dahulu", "dalam", "dan", "dapat",
"dari", "dari pada", "dekat", "demi", "demi kian",
"demi kian lah", "sedemi kian", "dengan", "depan",
"di", "di a", "di alah", "di ni", "diri", "dirinya",
"terdiri", "dong", "dulu", "ke", "enggak",
"enggaknya", "entah", "entahlah", "terhadap",
"terhadapnya", "hal", "hampir", "hanya",
"hanyalah", "harus", "haruslah", "harusnya",
"seharusnya", "hendak", "hendaklah", "hendaknya",
"hi ngga", "sehi ngga", "ia", "ialah", "ibarat",
"ingin", "inginkah", "inginkan", "ini", "inikah",
"ini lah", "itu", "itukah", "itu lah", "jangan",
"jangankan", "janganlah", "jika", "jikalau",
"juga", "justru", "kala", "kalau", "kalaulah",

```

"kal aupun", "kal i an", "kami", "kami l ah", "kamu",  
 "kamul ah", "kan", "kapan", "kapankah", "kapanpun",  
 "di karenakan", "karena", "karenanya", "keci l",  
 "kemudi an", "kenapa", "kepada", "kepadanya",  
 "keti ka", "seketi ka", "khususnya", "ki ni",  
 "ki ni l ah", "ki ranya", "seki ranya", "ki ta",  
 "ki tal ah", "kok", "l agi", "l agi an", "sel agi",  
 "l ah", "l ai n", "l ai nnya", "mel ai nkan", "sel aku",  
 "l al u", "mel al ui", "terl al u", "l ama", "l amanya",  
 "sel ama", "sel amanya", "l ebi h", "terl ebi h",  
 "bermacam", "macam", "semacam", "maka",  
 "makanya", "maki n", "mal ah", "mal ahan", "mampu",  
 "mampukah", "mana", "manakal a", "manal agi",  
 "masi h", "masi hkah", "semasi h", "masi ng",  
 "mau", "maupun", "semaunya", "memang", "mereka",  
 "merekal ah", "meski", "meski pun", "semul a",  
 "mungkin", "mungkin kah", "nah", "namun", "nant i",  
 "nant i nya", "nyari s", "ol eh", "ol ehnya",  
 "seorang", "seseorang", "pada", "padanya",  
 "padahal", "pal i ng", "sepanjang", "pantas",  
 "sepantasnya", "sepantasnyal ah", "para", "pasti",  
 "pasti l ah", "pernah", "pul a", "pun", "merupakan",  
 "rupanya", "serupa", "saat", "saatnya", "sesaat",  
 "saj a", "saj al ah", "sal i ng", "bersama", "sama",  
 "sesama", "sambi l", "sampai", "sana", "sangat",  
 "sangat l ah", "saya", "sayal ah", "sebab",  
 "sebabnya", "sebuah", "tersebut", "tersebut l ah",  
 "sedang", "sedangkan", "sedi ki t", "sedi ki tnya",  
 "segal a", "segal anya", "segera", "sesegera",  
 "sej ak", "sej enak", "sekal i", "sekal i an",  
 "sekal i pun", "sesekal i", "sekal i gus", "sekarang",  
 "seki tar", "seki tarnya", "sel a", "sel ai n",  
 "sel al u", "sel uruh", "sel uruhnya", "semaki n",  
 "sementara", "sempat", "semua", "semuanya",  
 "sendi ri", "sendi ri nya", "seol ah", "seperti",  
 "seperti nya", "seri ng", "seri ngnya", "serta",  
 "si apa", "si apakah", "si apapun", "di si ni",  
 "di si ni l ah", "si ni", "si ni l ah", "sesuatu",  
 "sesuatunya", "suatu", "sesudah", "sesudahnya",  
 "sudah", "sudahkah", "sudah l ah", "supaya", "tadi",  
 "tadi nya", "tak", "tanpa", "setel ah", "tel ah",  
 "tentang", "tentu", "tentul ah", "tentunya",  
 "tertentu", "seterusnya", "tapi", "tetapi",  
 "seti ap", "ti ap", "seti daknya", "ti dak",  
 "ti dakkah", "ti dakl ah", "toh", "waduh", "wah",  
 "wahai", "sewaktu", "wal au", "wal aupun", "wong",  
 "untuk", "yai tu", "yakni", "yang", "untuk",  
 "l ancara", "teri makasi h", "arah", "j al an", "l ight",  
 "surabaya", "pol i si", "i nfonya", "i nfo", "macet",  
 "hari", "tol ong", "padat"))

# tokenizing the corpus

```

dtm1 <- DocumentTermMatrix(clean_corpus) #Tf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtTfIdf)) #TfIdf
dtm1 <- DocumentTermMatrix(clean_corpus,
control=list(wei ghting=wei ghtBin)) #binary

# divide corpus into training and test data
train_size <- floor(0.7*nrow(bismlah))
set.seed(141321)
train_indices <- sample(seq_len(nrow(bismlah)),
size=train_size)
train_corpus <- clean_corpus[ train_indices]
test_corpus <- clean_corpus[-train_indices]
dtm1_train <- dtm1[train_indices, ]

#Create document-term matrices using frequent
words
twotter_train <- DocumentTermMatrix(train_corpus,
control=list(dictionary = "ramai", "good",
"terpantau", "lintas", "traffic", "kasih",
"semoga", "dolog", "kebakaran", "air",
"wonokromo", "alhamdulillah", "gubsuryo", "raya",
"mohon", "mobil", "motor", "kawan", "selamat",
"min", "daerah", "pagi", "jam", "pdam", "mati",
"lampu", "tolong", "jembatan", "unesa", "pasar"))
twotter_test <- DocumentTermMatrix(test_corpus,
control=list(dictionary = "ramai", "good",
"terpantau", "lintas", "traffic", "kasih",
"semoga", "dolog", "kebakaran", "air",
"wonokromo", "alhamdulillah", "gubsuryo", "raya",
"mohon", "mobil", "motor", "kawan", "selamat",
"min", "daerah", "pagi", "jam", "pdam", "mati",
"lampu", "tolong", "jembatan", "unesa", "pasar"))

#Convert count information to "Yes", "No"
convert_count <- function(x) {
  y <- ifelse(x > 0, 1, 0)
  y <- factor(y, levels=c(0, 1), labels=c("No",
"Yes"))
  y
}

#Convert document-term matrices
twotter_train <- apply(twotter_train, 2,
convert_count)
twotter_test <- apply(twotter_test, 2,
convert_count)

#the naive bayes
library(e1071)

```

```

twitter_classifier <- naiveBayes(twitter_train,
factor(bi_smi_lah$KLASIFIKASI[train_indices]))

#Evaluate the performance on the test data
predicted <- predict(twitter_classifier,
twitter_test)
actual <- bi_smi_lah$KLASIFIKASI[-train_indices]
100*sum(predicted==actual)/length(predicted)

```

## A.2 SVM

Penjelasan untuk cara Gridsearch yang dilakukan dengan mengubah nilai C dan gamma.

- C in seq(-5, -4, by=1) → nilai cost yang paling minimum sampai maksimum. By adalah jarak nilai cost yang diinginkan, apakah 0,1/0,001/1
- gamma in seq(-5, -4, by=1) → nilai gamma yang paling minimum sampai maksimum. By adalah jarak nilai gamma yang diinginkan, apakah 0,1/0,001/1

### A.2.1 Kode program model 1

```

e100 <- read.csv("~/cobanaivebayes/awal.csv", sep
= ";", colClasses = c("factor", "character"))
library(NLP)
library(tm)

# Split the data into a 70:30 training:testing
datasets
train_size <- floor(0.7*nrow(e100))
set.seed(141321)
train_indices <- sample(seq_len(nrow(e100)), size
= train_size)
train_data <- e100[train_indices, ]
test_data <- e100[-train_indices, ]
#lihatit teks
train_docs <- Corpus(VectorSource(train_data$text))

# Clean corpus
clean_docs <- tm_map(train_docs,
content_transformer(tolower))
clean_docs <- tm_map(train_docs, removeNumbers)

```

```

clean_docs <- tm_map(train_docs,
removePunctuation)
clean_docs <- tm_map(train_docs, stripWhitespace)

# Create dtm, pembobotan dengan Tf
dtm_train <- DocumentTermMatrix(train_docs)

# Transform dtm to matrix to data frame - df is
easier to work with
df_train <- as.data.frame(data.matrix(dtm_train),
stringsAsFactors = FALSE)

# names(df_train)
test_docs <- Corpus(VectorSource(test_data$text))

# Clean corpus test data
test_docs <- tm_map(test_docs,
content_transformer(tolower))
test_docs <- tm_map(test_docs, removeNumbers)
test_docs <- tm_map(test_docs, removePunctuation)
test_docs <- tm_map(test_docs, stripWhitespace)

dtm_test <- DocumentTermMatrix(test_docs, control
= list(dictionary = names(df_train)))
df_test <- as.data.frame(data.matrix(dtm_test),
stringsAsFactors = FALSE)

label_train <- train_data$label
label_test <- test_data$label

library(e1071)
#dengan kernel RBF
model <- svm(dtm_train, label_train)

#dengan kernel SVM
model <- svm(dtm_train, label_train, kernel =
"linear")

# Prediction on train data
predict_train <- predict(model, df_train)
table(predict_train, label_train)

```

```
# Prediction on test data
predict_test <- predict(model, df_test)
table(predict_test, label_test)

# Akurasi
100*sum(predict_test==label_test)/length(label_test)
```

### A.2.2 Kode program model 2

```
e100 <- read.csv("~/cobanavebayes/awal.csv", sep =
";", colClasses = c("factor", "character"))
library(NLP)
library(tm)

# Split the data into a 70:30 training:testing
datasets
train_size <- floor(0.7*nrow(e100))
set.seed(141321)
train_indices <- sample(seq_len(nrow(e100)), size =
train_size)
train_data <- e100[ train_indices, ]
test_data <- e100[-train_indices, ] #hati-hati teks
train_docs <- Corpus(VectorSource(train_data$text))

# Clean corpus
clean_docs <- tm_map(train_docs,
content_transformer(tolower))
clean_docs <- tm_map(train_docs, removeNumbers)
clean_docs <- tm_map(train_docs, removePunctuation)
clean_docs <- tm_map(train_docs, stripWhitespace)

# Create dtm, pembobotan dengan 3 metode
dtm_train <- DocumentTermMatrix(train_docs) #Tf
dtm_train <- DocumentTermMatrix(train_docs,
control=list(weighting=weightTfIdf)) #TfIdf
dtm_train <- DocumentTermMatrix(train_docs,
control=list(weighting=weightBin)) #binary

# Transform dtm to matrix to data frame - df is easier
to work with
df_train <- as.data.frame(data.matrix(dtm_train),
stringsAsFactors = FALSE)

# names(df_train)
```

```

test_docs <- Corpus(VectorSource(test_data$text))

# Clean corpus test data
test_docs <- tm_map(test_docs,
  content_transformer(tolower))
test_docs <- tm_map(test_docs, removeNumbers)
test_docs <- tm_map(test_docs, removePunctuation)
test_docs <- tm_map(test_docs, stripWhitespace)

dtm_test <- DocumentTermMatrix(test_docs, control =
  list(dictionary = names(df_train)))
df_test <- as.data.frame(data.matrix(dtm_test),
  stringsAsFactors = FALSE)

label_train <- train_data$label
label_test <- test_data$label

library(e1071)
#dengan kernel RBF
model <- svm(dtm_train, label_train)

#dengan kernel SVM
model <- svm(dtm_train, label_train, kernel =
  "linear")

# Prediction on train data
predict_train <- predict(model, df_train)
table(predict_train, label_train)

# Prediction on test data
predict_test <- predict(model, df_test)
table(predict_test, label_test)

# Akurasi
100*sum(predict_test==label_test)/length(label_test)

```

### A.2.3 Kode program model 3

```

e100 <- read.csv("~/cobanai vebayes/coba_svm.csv",
  sep = ";", colClasses = c("factor", "character"))
library(NLP)
library(tm)

```



```

# Split the data into a 70:30 training:testing
datasets
train_size <- floor(0.7*nrow(e100))
set.seed(141321)
train_indices <- sample(seq_len(nrow(e100)), size =
  train_size)
train_data <- e100[ train_indices, ]
test_data <- e100[-train_indices, ]
#lihat isi teks
train_docs <- Corpus(VectorSource(train_data$text))

# Clean corpus
clean_docs <- tm_map(train_docs,
  content_transformer(tolower))
clean_docs <- tm_map(train_docs, removeNumbers)
clean_docs <- tm_map(train_docs,
  removePunctuation)
clean_docs <- tm_map(train_docs, stripWhitespace)

# Create dtm, pembobotan
dtm_train <- DocumentTermMatrix(train_docs)

# Transform dtm to matrix to data frame - df is
easier to work with
df_train <- as.data.frame(data.matrix(dtm_train),
  stringsAsFactors = FALSE)

# names(df_train)
test_docs <- Corpus(VectorSource(test_data$text))

# Clean corpus test data
test_docs <- tm_map(test_docs,
  content_transformer(tolower))
test_docs <- tm_map(test_docs, removeNumbers)
test_docs <- tm_map(test_docs, removePunctuation)
test_docs <- tm_map(test_docs, stripWhitespace)

dtm_test <- DocumentTermMatrix(test_docs, control
  = list(dictionary = names(df_train)))
df_test <- as.data.frame(data.matrix(dtm_test),
  stringsAsFactors = FALSE)

label_train <- train_data$label
label_test <- test_data$label

```

```

library(e1071)
#dengan gridsearch
for(C in seq(-5, -4, by=1)){
  for(gamma in seq(-6, -4, by=1)){
    model <- svm(dtm_train, label_train, gamma =
2^gamma, cost = 2^C)
    predict_train <- predict(model, df_train)
    table(predict_train, label_train)

    predict_test <- predict(model, df_test)
    table(predict_test, label_test)

    print(100*sum(predict_test==label_test)/length(label_test))
  }
}

```

Gridsearch dilakukan dengan mengubah nilai C dan gamma.

- C in seq(-5, -4, by=1) → nilai cost yang paling minimum sampai maksimum. By adalah jarak nilai cost yang diinginkan, apakah 0,1/0,001/1
- gamma in seq(-5, -4, by=1) → nilai gamma yang paling minimum sampai maksimum. By adalah jarak nilai gamma yang diinginkan, apakah 0,1/0,001/1

#### A.2.4 Kode program model 4

```

e100 <- read.csv("~/cobanai vebayes/coba_svm.csv",
sep = ";", colClasses = c("factor", "character"))
library(NLP)
library(tm)

# Split the data into a 70:30 training:testing
datasets
train_size <- floor(0.7*nrow(e100))
set.seed(141321)
train_indices <- sample(seq_len(nrow(e100)), size =
train_size)
train_data <- e100[ train_indices, ]
test_data <- e100[-train_indices, ] #lihat isi teks
train_docs <- Corpus(VectorSource(train_data$text))

```

```

# Clean corpus
clean_docs <- tm_map(train_docs,
content_transformer(tolower))
clean_docs <- tm_map(train_docs, removeNumbers)
clean_docs <- tm_map(train_docs, removePunctuation)
clean_docs <- tm_map(train_docs, stripWhitespace)

# Create dtm, pembobotan
dtm_train <- DocumentTermMatrix(train_docs)

# Transform dtm to matrix to data frame - df is
easier to work with
df_train <- as.data.frame(data.matrix(dtm_train),
stringsAsFactors = FALSE)

# names(df_train)
test_docs <- Corpus(VectorSource(test_data$text))
# Clean corpus test data
test_docs <- tm_map(test_docs,
content_transformer(tolower))
test_docs <- tm_map(test_docs, removeNumbers)
test_docs <- tm_map(test_docs, removePunctuation)
test_docs <- tm_map(test_docs, stripWhitespace)

dtm_test <- DocumentTermMatrix(test_docs, control =
list(dictionary = names(df_train)))
df_test <- as.data.frame(data.matrix(dtm_test),
stringsAsFactors = FALSE)

label_train <- train_data$label
label_test <- test_data$label

library(e1071)
#dengan gridsearch
for(C in seq(-5, -4, by=1)){
  for(gamma in seq(-6, -4, by=1)){
    model <- svm(dtm_train, label_train, kernel =
"linear", gamma = 2^gamma, cost = 2^C)
    predict_train <- predict(model, df_train)
    table(predict_train, label_train)

    predict_test <- predict(model, df_test)
    table(predict_test, label_test)
  }
}

```

```

print(100*sum(predi ct_test==l abel _test)/l ength(l abe
l _test))
}
}

```

### A.2.5 Kode program model 5

```

e100 <- read.csv("~/cobanaivebayes/coba_svm.csv",
sep = ";", colClasses = c("factor", "character"))
library(NLP)
library(tm)

# Split the data into a 70:30 training:testing
# datasets
train_size <- floor(0.7*nrow(e100))
set.seed(141321)
train_indices <- sample(seq_len(nrow(e100)), size =
train_size)
train_data <- e100[ train_indices, ]
test_data <- e100[-train_indices, ]
#lihatisi teks
train_docs <-
Corpus(VectorSource(train_data$text))

# Clean corpus
clean_docs <- tm_map(train_docs,
content_transformer(tolower))
clean_docs <- tm_map(train_docs, removeNumbers)
clean_docs <- tm_map(train_docs, removePunctuation)
clean_docs <- tm_map(train_docs, stripWhitespace)

# Create dtm, 3 pembobotan
dtm_train <- DocumentTermMatrix(train_docs)
dtm_train <- DocumentTermMatrix(train_docs,
control=list(weighing=weightTfIdf)) #Tf idf
dtm_train <- DocumentTermMatrix(train_docs,
control=list(weighing=weightBin)) #binary

# Transform dtm to matrix to data frame - df is
# easier to work with
df_train <- as.data.frame(data.matrix(dtm_train),
stringsAsFactors = FALSE)

```

```

# names(df_train)
test_docs <- Corpus(VectorSource(test_data$text))

# Clean corpus test data
test_docs <- tm_map(test_docs,
  content_transformer(tolower))
test_docs <- tm_map(test_docs, removeNumbers)
test_docs <- tm_map(test_docs, removePunctuation)
test_docs <- tm_map(test_docs, stripWhitespace)

dtm_test <- DocumentTermMatrix(test_docs, control =
  list(dictionary = names(df_train)))
df_test <- as.data.frame(data.matrix(dtm_test),
  stringsAsFactors = FALSE)

label_train <- train_data$label
label_test <- test_data$label

library(e1071)
#dengan gridsearch
for(C in seq(-5, -4, by=1)){
  for(gamma in seq(-6, -4, by=1)){
    model <- svm(dtm_train, label_train, gamma =
      2^gamma, cost = 2^C)
    predict_train <- predict(model, df_train)
    table(predict_train, label_train)

    predict_test <- predict(model, df_test)
    table(predict_test, label_test)

    print(100*sum(predict_test==label_test)/length(label_test))
  }
}

```

### A.2.6 Kode program model 6

```

e100 <- read.csv("~/cobanavebayes/coba_svm.csv",
  sep = ";", colClasses = c("factor", "character"))
library(NLP)
library(tm)

# Split the data into a 70:30 training/testing
  datasets

```

```

train_size <- floor(0.7*nrow(e100))
set.seed(141321)
train_indices <- sample(seq_len(nrow(e100)), size =
train_size)
train_data <- e100[ train_indices, ]
test_data <- e100[ -train_indices, ]
#I i hati si teks
train_docs <- Corpus(VectorSource(train_data$text))

# Clean corpus
clean_docs <- tm_map(train_docs,
content_transformer(tolower))
clean_docs <- tm_map(train_docs, removeNumbers)
clean_docs <- tm_map(train_docs, removePunctuation)
clean_docs <- tm_map(train_docs, stripWhitespace)

# Create dtm, 3 pembobotan
dtm_train <- DocumentTermMatrix(train_docs)
dtm_train <- DocumentTermMatrix(train_docs,
control=list(weighing=weightTfIdf)) #Tf idf
dtm_train <- DocumentTermMatrix(train_docs,
control=list(weighing=weightBin)) #binary

# Transform dtm to matrix to data frame - df is
easier to work with
df_train <- as.data.frame(data.matrix(dtm_train),
stringsAsFactors = FALSE)

# names(df_train)
test_docs <- Corpus(VectorSource(test_data$text))

# Clean corpus test data
test_docs <- tm_map(test_docs,
content_transformer(tolower))
test_docs <- tm_map(test_docs, removeNumbers)
test_docs <- tm_map(test_docs, removePunctuation)
test_docs <- tm_map(test_docs, stripWhitespace)

dtm_test <- DocumentTermMatrix(test_docs, control =
list(dictionary = names(df_train)))
df_test <- as.data.frame(data.matrix(dtm_test),
stringsAsFactors = FALSE)

```

```

label_train <- train_data$label
label_test  <- test_data$label

library(e1071)
#dengan gridsearch
for(C in seq(-5, -4, by=1)){
  for(gamma in seq(-6, -4, by=1)){
    model <- svm(dtm_train, label_train, kernel =
"linear", gamma = 2^gamma, cost = 2^C)
    predict_train <- predict(model, df_train)
    table(predict_train, label_train)

    predict_test <- predict(model, df_test)
    table(predict_test, label_test)

    print(100*sum(predict_test==label_test)/length(label_test))
  }
}

```

## A.3 SHINY

### A.3.1. Uir

```

shinyUI(fluidPage(

  # Application title
  titlePanel("Visualisasi Analisis Sentimen
Terhadap Kinerja Pelayanan Publik Di Kota Surabaya
Berdasarkan Klasiifikasi Komentar Di Media Sosial
Dengan Menggunakan Algoritma Naïve Bayes"),

  # Show Word Cloud
  mainPanel (
    tabsetPanel(type = c("pills"),
      tabPanel("Wordcloud Semua Data",
        sidebarPanel (
          selectInput("klasiifikasi",
            "Pilih kelas klasiifikasi : ",
            klasiifikasi =
            klasiifikasi),
            actionButton("update",
            "Change"),
            hr(),

```

```

        sliderInput("freq", "Mi ni mum
Frequency: ",
                    min = 1,
                    max = 100, value = 5),
        sliderInput("max", "Maxi mum
Number of Words: ",
                    min = 1,
                    max = 200, value = 20)
    ),
    mainPanel (
        plotOutput("plot"),
        h5('Vi sua li sa si Wordcloud pada
tab ini memuat informasi tentang kata-kata yang
sering muncul di masi ng-masi ng kelas.
Kel as kl asi fi ka si yang
di tam pi lkan yai tu posi ti f, negati f, dan netral.
Pengaturan mi ni mum dan
maksimum kata juga bi sa di atur
dengan mengub ah Mi ni mum
Frequency dan Maxi mum Number of Words'))
    ),
    tabPanel("Wordcloud Data Surabaya",
        sidebarPanel (
            selectInput("kel askl asi fi ka si ",
"Pi li h kel as kl asi fi ka si : ",
                        choi ces =
kel askl asi fi ka si ),
            acti onButton("update1", "Change"),
            hr(),
            sliderInput("freq", "Mi ni mum
Frequency: ",
                        min = 1, max = 100,
                        value = 5),
            sliderInput("max", "Maxi mum Number
of Words: ",
                        min = 1, max = 200,
                        value = 20)
        ),
        mainPanel (
            plotOutput("plot3"),
            h5('Vi sua li sa si Wordcloud pada tab
ini memuat informasi tentang kata-kata yang seri ng
muncul di masi ng-masi ng kelas.

```



```

        Kelas     klasi fi kasi     yang
di tampilkan yai tu posi ti f, negati f, dan netral .
        Pengaturan     mi ni mum     dan
maksimum kata juga bisa diatur
        dengan     mengubah     Mi ni mum
Frequency dan Maxi mum Number of Words' ))
    ),
    tabPanel ("Streamgraph     Semua     Data",
streamgraphOutput("pl ot1"),
    h5('Vi sua li sa si i ni menampi lkan jum lah tweet
dari masi ng-masi ng kel as pada seti ap hari nya' )),

    tabPanel ("Streamgraph     Data     Surabaya",
streamgraphOutput("pl ot2"),
    h5('Vi sua li sa si i ni menampi lkan
jum lah tweet dari masi ng-masi ng kel as pada seti ap
hari nya' ))
    )
)
)
)

```

### A.3.2. Global.R

```

library(NLP)
library(tm)
library(wordcloud)
library(RColorBrewer)
library(memoise)
library(dplyr)
library(streamgraph)
library(stats)
library(base)

# The list of valid books
klasi fi kasi <- list("Negati f" = "negati fsemua",
"Posi ti f" = "posi ti fsemua", "Netral" =
"netral semua")
# Using "memoise" to automatically cache the
results
getTermMatrix <- memoise(function(klasi fi kasi ) {
  # Careful not to let just any name slip in here;
a
  # malicious user could manipulate this value.
  if (! (klasi fi kasi %i n% klasi fi kasi ))

```

```

stop("Unknown klasi fi kasi ")

text      <-      readLines(sprintf("%. /%s. txt. gz",
klasi fi kasi ),
                                encodi ng="UTF-8")
docs <- Corpus(VectorSource(text))

toSpace <- content_transformer(function (x ,
pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/" )
docs <- tm_map(docs, toSpace, "@" )
docs <- tm_map(docs, toSpace, "\\|")
docs      <-      tm_map(docs,
content_transformer(tolower))
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, stripWhitespace)
docs <- tm_map(docs, removeWords, c("ada" ,
"adanya" , "adal ah" , "adapun" , "agak" , "agak nya"
, "agar" , "akan" , "akankah" , "akhi r nya" , "aku"
, "akul ah" , , "amat" , , "amatl ah" , , "anda" ,
"andal ah" ,
"antar" , "di antaranya" , "antara" , "antaranya"
, "di antara" , "apa" , "apaan" , "mengapa" ,
"apabi la" , , "apakah" , "apal agi " , "apatah" ,
"atau" , "ataukah" ,
"ataupun" , "bagai " , "bagai kan" , "sebagai " ,
"sebagai nya" , , "bagai mana" , , "bagai manapun" ,
"sebagai mana" , "bagai manakah" , "bagi " , "bahkan"
, "bahwa" ,
"bahwasanya" , , "sebal i knya" , , "banyak" ,
"sebanyak" , "beberapa" , "seberapa" , "begi ni " ,
"begi ni an" , , "begi ni kah" , , "begi ni l ah" ,
"sebegi ni " , "begi tu" ,
"begi tukah" , , "begi tul ah" , , "begi tupun" ,
"sebegi tu" , , "bel um" , , "bel uml ah" , , "sebel um" ,
"sebel um nya" , , "sebenarnya" , , "berapa" ,
"berapakah" , "berapal ah" , "berapapun" ,
"betul kah" , "sebetul nya" , "bi asa" , "bi asanya" ,
"bi l a" , "bi l akah" , "bi sa" , "bi sakah" , "sebi sanya" ,
"bol eh" , "bol ehkah" , "bol ehl ah" , "buat" , "bukan" ,
"bukankah" ,
"bukantl ah" , , "bukannya" , , "cuma" , "percuma" ,
"dahul u" , , "dal am" , , "dan" , , "dapat" , , "dari " ,

```

"dari pada", "dekat", "demi", "demi ki an",  
 "demi ki anl ah", "sedemi ki an",  
 "dengan", "depan", "di", "di a", "di al ah",  
 "di ni", "di ri", "di ri nya", "terdi ri", "dong",  
 "dul u", "ke", "enggak", "enggaknya", "entah",  
 "entah ah", "terhadap", "terhadapnya",  
 "hal", "hampi r", "hanya", "hanyal ah", "harus",  
 "harusl ah", "harusnya", "seharusnya", "hendak",  
 "hendakl ah", "hendaknya", "hi ngga", "sehi ngga",  
 "i a", "i al ah", "i barat",  
 "i ngi n", "i ngi nkah", "i ngi nkan", "i ni",  
 "i ni kah", "i ni l ah", "i tu", "i tukah", "i tul ah",  
 "j angan", "j angankan", "j anganl ah", "j i ka",  
 "j i kal au", "j uga", "j ustru", "kal a",  
 "kal au", "kal aul ah", "kal aupun", "kal i an",  
 "kami", "kami l ah", "kamu", "kamul ah", "kan",  
 "kapan", "kapankah", "kapanpun", "di karenakan",  
 "karena", "karenanya", "keci l",  
 "kemudi an", "kenapa", "kepada", "kepadanya",  
 "keti ka", "seketi ka", "khususnya", "ki ni",  
 "ki ni l ah", "ki ranya", "seki ranya", "ki ta",  
 "ki tal ah", "kok", "l agi", "l agi an",  
 "sel agi", "l ah", "l ai n", "l ai nnya", "mel ai nkan",  
 "sel aku", "l al u", "mel al ui", "terl al u", "l ama",  
 "l amanya", "sel ama", "sel amanya", "l ebi h",  
 "terl ebi h", "bermacam",  
 "macam", "semacam", "maka", "makanya",  
 "maki n", "mal ah", "mal ahan", "mampu", "mampukah",  
 "mana", "manakal a", "manal agi", "masi h",  
 "masi hkah", "semasi h", "masi ng",  
 "mau", "maupun", "semaunya", "memang", "mereka",  
 "merekal ah", "meski", "meski pun", "semul a",  
 "mungkin", "mungkin kah", "nah", "namun", "nanti",  
 "nanti nya", "nyari s", "ol eh", "ol ehnya",  
 "seorang",  
 "seseorang", "pada", "padanya", "padahal",  
 "pal ing", "sepanj ang", "pantas", "sepantasnya",  
 "sepantasnyal ah", "para", "pasti", "pasti l ah",  
 "pernah", "pul a", "pun",  
 "merupakan", "rupanya", "serupa", "saat",  
 "saatnya", "sesaat", "saj a", "saj al ah", "sal i ng",  
 "bersama", "sama", "sesama", "sambi l", "sampai",  
 "sana", "sangat",  
 "sangatl ah", "saya", "sayal ah", "sebab",  
 "sebabnya", "sebuah", "tersebut", "tersebutl ah",

```
"sedang", "sedangkan", "sedi ki t", "sedi ki tnya",
"segal a", "segal anya",
"segera", "sesegera", "sej ak", "sej enak",
"sekali i", "sekali an", "sekali pun", "sesekali i",
"sekali gus", "sekarang", "seki tar", "seki tarnya",
"sel a", "sel ai n",
"sel al u", "sel uruh", "sel uruhnya", "semaki n",
"sementara", "sempat", "semua", "semuanya",
"sendi ri", "sendi ri nya", "seol ah", "seperti",
"seperti nya", "seri ng",
"seri ngnya", "serta", "si apa", "si apakah",
"si apapun", "di si ni", "di si ni lah", "si ni",
"si ni lah", "sesuatu", "sesuatunya", "suatu",
"sesudah", "sesudahnya",
"sudah", "sudahkah", "sudah lah", "supaya",
"tadi", "tadi nya", "tak", "tanpa", "setel ah",
"tel ah", "tentang", "tentu", "tentul ah",
"tentunya", "tertentu",
"seterusnya", "tapi", "tetapi", "seti ap",
"ti ap", "seti daknya", "ti dak", "ti dakkah",
"ti dakl ah", "toh", "waduh", "wah", "wahi",
"sewaktu", "wal au", "wal aupun",
"wong", "yai tu", "yakni", "yang", "untuk"))
```

```
dtm <- TermDocumentMatrix(docs, control =
list(minWordLength = 1))
```

```
m = as.matrix(dtm)
```

```
sort(rowSums(m), decreasing = TRUE)
})
```

```
# Data Surabaya #
```

```
#The list of valid books
```

```
kel askl asi fi kasi <- list("Negatif" =
"negati fsby", "Posi ti f" = "posi ti fsby", "Netral"
= "netral sbby")
```

```
# Using "memoi se" to automaticall y cache the
resul ts
```

```
getTermMatrix
memoi se(function(kel askl asi fi kasi ) {
```

```

# Careful not to let just any name slip in here;
# a malicious user could manipulate this value.
if (! (kel askl asi fi kasi %i n% kel askl asi fi kasi ))
  stop("Unknown kel askl asi fi kasi ")

text1      <-      readLines(sprintf("%. /%s. txt. gz",
kel askl asi fi kasi ),
                        encoding="UTF-8")
docs1 <- Corpus(VectorSource(text1))

toSpace1 <- content_transformer(function (x ,
pattern ) gsub(pattern, " ", x))
docs1 <- tm_map(docs1, toSpace1, "/" )
docs1 <- tm_map(docs1, toSpace1, "@")
docs1 <- tm_map(docs1, toSpace1, "\\|")
docs1      <-      tm_map(docs1,
content_transformer(tolower))
docs1 <- tm_map(docs1, removeNumbers)
docs1 <- tm_map(docs1, removePunctuation)
docs1 <- tm_map(docs1, stripWhitespace)
docs1 <- tm_map(docs1, removeWords, c("ada" ,
"adanya" , "adalah" , "adapun" , "agak" , "agak nya"
, "agar" , "akan" , "akankah" , "akhir nya" , "aku"
, "akulah" , "amat" , "amat lah" , "anda" ,
"andal ah" ,
"antar" ,
"di antaranya" , "antara" , "antaranya" ,
"di antara" , "apa" , "apaan" , "mengapa" ,
"apabila" , "apakah" , "apalagi" , "apatah" ,
"atau" , "ataukah" ,
"ataupun" ,
"bagai" , "bagai kan" , "sebagai" , "sebagai nya" ,
"bagai mana" , "bagai manapun" , "sebagai mana" ,
"bagai manakah" , "bagi" , "bahkan" , "bahwa" ,
"bahwasanya" ,
"sebal i nya" , "banyak" , "sebanyak" , "beberapa"
, "seberapa" , "begi ni" , "begi ni an" , "begi ni kah"
, "begi ni lah" , "sebegi ni" , "begi tu" ,
"begi tukah" ,
"begi tul ah" , "begi tupun" , "sebegi tu" , "bel um" ,
"bel um lah" , "sebel um" , "sebel um nya" ,
"sebenarnya" , "berapa" , "berapakah" , "berapal ah" ,
"berapapun" ,
"betul kah" ,
"sebetul nya" , "bi asa" , "bi asanya" , "bi la" ,

```

"bi l akah", "bi sa", "bi sakah", "sebi sanya",  
 "bol eh", "bol ehkah", "bol ehl ah", "buat", "bukan",  
 "bukankah",  
 "bukanl ah",  
 "bukannya", "cuma", "percuma", "dahul u", "dal am",  
 "dan", "dapat", "dari ", "dari pada", "dekat",  
 "demi ", "demi ki an", "demi ki anl ah", "sedemi ki an",  
 "dengan",  
 "depan", "di ", "di a", "di al ah", "di ni ", "di ri ",  
 "di ri nya", "terdi ri ", "dong", "dul u", "ke",  
 "enggak", "enggaknya", "entah", "entahl ah",  
 "terhadap", "terhadapnya",  
 "hal ",  
 "hampi r", "hanya", "hanyal ah", "harus",  
 "harusl ah", "harusnya", "seharusnya", "hendak",  
 "hendakl ah", "hendaknya", "hi ngga", "sehi ngga",  
 "i a", "i al ah", "i barat",  
 "i ngi n",  
 "i ngi nkah", "i ngi nkan", "i ni ", "i ni kah", "i ni l ah",  
 "i tu", "i tukah", "i tul ah", "j angan", "j angankan",  
 "j anganl ah", "j i ka", "j i kal au", "j uga", "j ustru",  
 "kal a",  
 "kal au",  
 "kal aul ah", "kal aupun", "kal i an", "kami ",  
 "kami l ah", "kamu", "kamul ah", "kan", "kapan",  
 "kapankah", "kapanpun", "di karenakan", "karena",  
 "karenanya", "keci l",  
 "kemudi an",  
 "kenapa", "kepada", "kepadanya", "keti ka",  
 "seketi ka", "khususnya", "ki ni ", "ki ni l ah",  
 "ki ranya", "seki ranya", "ki ta", "ki tal ah", "kok",  
 "l agi ", "l agi an",  
 "sel agi ",  
 "l ah", "l ai n", "l ai nnya", "mel ai nkan", "sel aku",  
 "l al u", "mel al ui ", "terl al u", "l ama", "l amanya",  
 "sel ama", "sel amanya", "l ebi h", "terl ebi h",  
 "bermacam",  
 "macam",  
 "semacam", "maka", "makanya", "maki n", "mal ah",  
 "mal ahan", "mampu", "mampukah", "mana",  
 "manakal a", "manal agi ", "masi h", "masi hkah",  
 "semasi h", "masi ng",  
 "mau",  
 "maupun", "semaunya", "memang", "mereka",

"mereka lah", "meski", "meskipun", "semula",  
 "mungkin", "mungkinlah", "nah", "namun", "nantinya",  
 "nantinya", "nyaris", "oleh", "olehnya",  
 "seorang",  
 "seseorang",  
 "pada", "padanya", "padahal", "paling",  
 "sepanjang", "pantas", "sepantasnya",  
 "sepantasnyalah", "para", "pasti", "pastilah",  
 "pernah", "pula", "pun",  
 "merupakan",  
 "rupanya", "serupa", "saat", "saatnya", "sesaat",  
 "saja", "sajalah", "salah", "bersama", "sama",  
 "sesama", "sambil", "sampai", "sana", "sangat",  
 "sangatlah",  
 "saya", "sayalah", "sebab", "sebabnya", "sebuah",  
 "tersebut", "tersebutlah", "sedang", "sedangkan",  
 "sedikit", "sedikitnya", "segala", "segalanya",  
 "segera",  
 "sesegera", "sejak", "sejenak", "sekali",  
 "sekali", "sekali", "sekalipun", "sekalih", "sekaligus",  
 "sekarang", "sekitar", "sekitarnya", "selah",  
 "selain",  
 "selalu",  
 "seluruh", "seluruhnya", "semakin", "sementara",  
 "sempat", "semua", "semuanya", "sendiri",  
 "sendirinya", "seolah", "seperti", "sepertinya",  
 "sering",  
 "seringnya",  
 "serta", "siapa", "siapa", "siapa", "siapa",  
 "di sini", "di sini", "di sini", "di sini",  
 "sesuatu", "sesuatunya", "suatu", "sesudah",  
 "sesudahnya",  
 "sudah",  
 "sudahkah", "sudahlah", "supaya", "tadi",  
 "tadinya", "tak", "tanpa", "setelah", "telah",  
 "tentang", "tentu", "tentulah", "tentunya",  
 "tertentu",  
 "seterusnya",  
 "tapi", "tetapi", "setiap", "tiap", "setidaknya",  
 "tidak", "tidak", "tidak", "tidak",  
 "wah", "wahai", "sewaktu", "waktu", "walaupun",  
 "wong",  
 "yaitu", "yakni", "yang", "untuk"))

```

dtm1 <- TermDocumentMatrix(docs1, control =
list(minWordLength = 1))

m1 = as.matrix(dtm1)

sort(rowSums(m1), decreasing = TRUE)
}
)

```

### A.3.3. Server.R

```

library(shiny)
library(stats)
library(base)
function(input, output, session) {

  terms <- reactive({
    # Change when the "update" button is
    pressed...
    input$update
    # ...but not for anything else
    isolate({
      withProgress({
        setProgress(message = "Processing
        corpus...")
        getTermMatrix(input$klasi_fikasi)
      })
    })
  })

  # Make the wordcloud drawing predictable during
  a session
  wordcloud_rep <- repeatabl e(wordcloud)
  output$plot <- renderPlot({
    v <- terms()
    wordcloud_rep(names(v), v, scale=c(4, 0.5),
    min.freq = input$freq,
    max.words=input$max,
    colors=brewer.pal (8,
    "Dark2"))))

  terms1 <- reactive({
    # Change when the "update" button is
    pressed...

```



```

input$update1
# ...but not for anything else
isolate({
  withProgress({
    setProgress(message = "Processing
corpus...")
    getTermMatrix(input$kelaskifiasi)
  })
})

#kelas wordcloud 1
wordcloud_rep1 <- repeatably(wordcloud)
output$plot3 <- renderPlot({
  v1 <- terms1()
  wordcloud_rep1(names(v1), v1, scale=c(4, 0.5),
    min.freq = input$freq,
max.words=input$max,
    colors=brewer.pal(8,
"Dark2"))})

colors <- brewer.pal(4, "PiYG")
output$plot1 <- renderStreamgraph({
  nyobagraph %>%
    mutate(tanggal =as.Date(tanggal ,
format="%d/%m/%y")) %>%
    group_by(tanggal , klaskifiasi) %>%
    streamgraph("klaskifiasi ", "jumlah",
"tanggal ") %>%
    sg_fill_manual(colors) %>%
    sg_axis_x("tanggal ") %>%
    sg_legend(show=TRUE, label="Klaskifiasi : ")
})

colors1 <- brewer.pal(3, "RdYlBu")
output$plot2 <- renderStreamgraph({
  graphsby %>%
    mutate(tanggal =as.Date(tanggal ,
format="%d/%m/%y")) %>%
    group_by(tanggal , klaskifiasi) %>%
    streamgraph("klaskifiasi ", "jumlah",
"tanggal ") %>%
    sg_fill_manual(colors1) %>%

```

```
sg_axis_x("tanggal ") %>%  
sg_legend(show=TRUE, label = "Klasi fikasi : ")  
})  
}
```

**LAMPIRAN B**  
**DAFTAR KATA WORDCLOUD**

**B.1     Positif**

**B.1.1   10**



A word cloud for B.1.1. The word 'lancar' is the largest and most prominent. Other words include 'jalan', 'good', 'lintas', 'terimakasih', 'ramai', 'terpantau', and 'arah'.

**B.1.2   20**



A word cloud for B.1.2. The words 'ramai' and 'lancar' are the most prominent.

**B.1.3   30, 40**



A word cloud for B.1.3. The word 'lancar' is the most prominent.

B.2 Negatif

B.2.1 40

A word cloud for B.2.1. The words are arranged in a vertical stack. At the top is 'surabaya', followed by 'polisi' and 'padat'. The word 'macet' is the largest and most prominent in the center. Below it is 'hari', then 'jalan', and finally 'arah' at the bottom, oriented vertically.

B.2.2 50

A word cloud for B.2.2. The words are arranged in a vertical stack. At the top is 'macet', followed by 'jalan'. Below 'jalan' is 'arah', then 'padat', 'surabaya', and 'polisi' at the bottom.

B.2.3 60

jalan  
arah  
macet  
padat

B.2.4 70, 80, 90

arah  
jalan  
macet

B.2.5 100

macet  
jalan

B.2.6 110

jalan

### B.3 Netral

**B.3.1 40**



B.3.2 60



B.3.3 80



B.3.4 100



A word cloud where the words are arranged in a roughly triangular shape. The word 'jalan' is the largest and at the top. Below it is 'terimakasih'. To the left is 'macet' and to the right is 'arah'. Below 'macet' is 'polisi'. Below 'arah' is 'hariinfo'. Below 'polisi' is 'mobil'. Below 'mobil' is 'motor'. Below 'motor' is 'mohon'. To the right of 'mohon' is 'tolong'. At the bottom is 'surabaya'.

jalan  
terimakasih  
macet arah  
polisi hariinfo  
mobil  
motor  
mohon tolong  
surabaya

B.3.5 120



A word cloud where the words are arranged in a roughly triangular shape. The word 'surabaya' is the largest and at the top. Below it is 'mohon'. To the left is 'macet' and to the right is 'arah'. Below 'macet' is 'jalan'. Below 'mohon' is 'arah'. Below 'arah' is 'jalan'.

terimakasih  
surabaya  
mohon  
macet arah  
jalan



surabaya  
jalan  
arah<sup>mohon</sup>  
macet

*Halaman ini sengaja dikosongkan*

## LAMPIRAN C

### DAFTAR KATA STOPWORD

#### C.1 Stopword dari github

ada	hampir	olehnya	tidaklah
adanya	hanya	seorang	toh
adalah	hanyalah	seseorang	waduh
adapun	harus	pada	wah
agak	haruslah	padanya	wahai
agaknya	harusnya	padahal	sewaktu
agar	seharusnya	paling	walaupun
akan	hendak	sepanjang	walaupun
akankah	hendaklah	pantas	wong
akhirnya	hendaknya	sepantasnya	yaitu
aku	hingga	sepantasnyalah	yakni
akulah	sehingga	para	yang
amat	ia	pasti	tadi
amatlah	ialah	pastilah	tadinya
anda	ibarat	per	tak
andalah	ingin	pernah	tanpa
antar	inginkah	pula	setelah
diantaranya	inginkan	pun	telah
antaranya	inikah	rupanya	tentu
diantara	inilah	serupa	tentulah
apa	itu	saat	tentunya
apaan	itukah	saatnya	tertentu
mengapa	itulah	sesaat	seterusnya
apabila	jangan	saja	tapi
apakah	jangankan	sajalah	tetapi
apalagi	janganlah	saling	setiap
apatah	jika	bersama	tiap

atau	jikalau	sama	setidaknya
ataukah	juga	sesama	tidak
ataupun	justru	sambil	tidakkah
bagai	kala	sampai	semasih
bagaikan	kalau	sana	masing
sebagai	kalaulah	sangat	mau
sebagainya	kalaupun	sangatlah	maupun
bagaimana	kalian	saya	semaunya
bagaimanapun	kami	sayalah	memang
sebagaimana	kamilah	se	mereka
bagaimanakah	kamu	sebab	merekalah
bagi	kamulah	sebabnya	meski
bahkan	kan	sebuah	meskipun
bahwa	kapan	tersebut	semula
sebaliknya	kapanpun	sedang	mungkinkah
sebanyak	karena	sedikit	namun
seberapa	ke	segala	nantinya
beginian	kemudian	segera	oleh
beginikah	kenapa	sesegera	sedemikian
beginilah	kepada	sejak	dengan
sebegini	kepadanya	sejenak	depan
begitu	ketika	sekali	di
begitukah	seketika	sekalian	dia
begitulah	khususnya	sekalipun	dialah
begitupun	kini	sese kali	dini
sebegitu	kinilah	sekaligus	diri
belum	kiranya	sekarang	dirinya
belum lah	sekiranya	sekarang	terdiri
sebelum	kita	sekitar	dong

sebelumnya	kitalah	sekitarnya	dulu
sebenarnya	kok	sela	enggak
berapa	lagi	selain	enggaknya
berapakah	lagian	selalu	entah
berapalah	selagi	seluruh	entahlah
berapapun	lah	seluruhnya	terhadap
betulkah	lain	semakin	terhadapnya
sebetulnya	lainnya	sementara	hal
biasa	melainkan	sempat	dahulu
biasanya	selaku	semua	dalam
bila	lalu	semuanya	dan
bilakah	melalui	sendiri	dapat
bisa	terlalu	sendirinya	dari
bisakah	lama	seolah	daripada
sebisanya	lamanya	seperti	dekat
boleh	selama	sepertinya	demi
bolehkah	selama	sering	demikian
bolehlah	selamanya	seringnya	demikianlah
buat	lebih	serta	makin
bukan	terlebih	siapa	malah
bukankah	bermacam	siapakah	malahan
bukanlah	macam	siapapun	mampu
bukannya	semacam	disini	mampukah
cuma	maka	disinilah	mana
percuma	makanya	sini	manakala
sesuatunya	sesudahnya	sinilah	manalagi
suatu	sudah	sesuatu	masih
masihkah	sudahlah	supaya	

## C.2 Stopword model 6 skenario 1

"ada", "adanya", "adalah", "adapun", "agak", "agaknya", "agar", "akan", "akankah", "akhirnya", "aku", "akulah", "amat", "amatlah", "anda", "andalah", "antar", "diantaranya", "antara", "antaranya", "diantara", "apa", "apaan", "mengapa", "apabila", "apakah", "apalagi", "apatah", "atau", "ataukah", "ataupun", "bagai", "bagaikan", "sebagai", "sebagainya", "bagaimana", "bagaimanapun", "sebagaimana", "bagaimanakah", "bagi", "bahkan", "bahwa", "bahwasanya", "sebaliknya", "banyak", "sebanyak", "beberapa", "seberapa", "begini", "beginian", "beginikah", "beginilah", "sebegini", "begitu", "begitukah", "begitulah", "begitupun", "sebegitu", "belum", "belumlah", "sebelum", "sebelumnya", "sebenarnya", "berapa", "berapakah", "berapalah", "berapapun", "betulkah", "sebetulnya", "biasa", "biasanya", "bila", "bilakah", "bisa", "bisakah", "sebisanya", "boleh", "bolehkah", "bolehlah", "buat", "bukan", "bukankah", "bukanlah", "bukannya", "cuma", "percuma", "dahulu", "dalam", "dan", "dapat", "dari", "daripada", "dekat", "demi", "demikian", "demikianlah", "sedemikian", "dengan", "depan", "di", "dia", "dialah", "dini", "diri", "dirinya", "terdiri", "dong", "dulu", "ke", "enggak", "enggaknya", "entah", "entahlah", "terhadap", "terhadapnya", "hal", "hampir", "hanya", "hanyalah", "harus", "haruslah", "harusnya", "seharusnya", "hendak", "hendaklah", "hendaknya", "hingga", "sehingga", "ia", "ialah", "ibarat", "ingin", "inginkan", "inginkan", "ini", "inikah", "inilah", "itu", "itukah", "itulah", "jangan", "janganlah", "janganlah", "jika", "jikalau", "juga", "justru", "kala", "kalau", "kalaulah", "kalaupun", "kalian", "kami", "kamilah", "kamu", "kamulah", "kan", "kapan", "kapankah", "kapanpun", "dikarenakan", "karena", "karenanya", "kecil", "kemudian", "kenapa", "kepada", "kepadanya", "ketika", "seketika", "khususnya", "kini", "kinilah", "kiranya", "sekiranya", "kita", "kitalah", "kok", "lagi", "lagian",

"selagi", "lah", "lain", "lainnya", "melainkan", "selaku",  
 "lalu", "melalui", "terlalu", "lama", "lamanya", "selama",  
 "selamanya", "lebih", "terlebih", "bermacam",  
 "macam", "semacam", "maka", "makanya", "makin",  
 "malah", "malahan", "mampu", "mampukah", "mana",  
 "manakala", "manalagi", "masih", "masihkah", "semasih",  
 "masing", "mau", "maupun", "semaunya", "memang",  
 "mereka", "merekalah", "meski", "meskipun", "semula",  
 "mungkin", "mungkinkah", "nah", "namun", "nanti",  
 "nantinya", "nyaris", "oleh", "olehnya", "seorang",  
 "seseorang", "pada", "padanya", "padahal", "paling",  
 "sepanjang", "pantas", "sepantasnya", "sepantasnyalah",  
 "para", "pasti", "pastilah", "pernah", "pula", "pun",  
 "merupakan", "rupanya", "serupa", "saat", "saatnya",  
 "sesaat", "saja", "sajalah", "saling", "bersama", "sama",  
 "sesama", "sambil", "sampai", "sana", "sangat",  
 "sangatlah", "saya", "sayalah", "sebab", "sebabnya",  
 "sebuah", "tersebut", "tersebutlah", "sedang", "sedangkan",  
 "sedikit", "sedikitnya", "segala", "segalanya",  
 "segera", "sesegera", "sejak", "sejenak", "sekali", "sekalian",  
 "sekalipun", "sesekali", "sekaligus", "sekarang", "sekitar",  
 "sekitarnya", "sela", "selain", "selalu", "seluruh",  
 "seluruhnya", "semakin", "sementara", "sempat", "semua",  
 "semuanya", "sendiri", "sendirinya", "seolah", "seperti",  
 "sepertinya", "sering", "seringnya", "serta", "siapa",  
 "siapakah", "siapapun", "disini", "disinilah", "sini",  
 "sinilah", "sesuatu", "sesuatunya", "suatu", "sesudah",  
 "sesudahnya", "sudah", "sudahkah", "sudahlah", "supaya",  
 "tadi", "tadinya", "tak", "tanpa", "setelah", "telah", "tentang",  
 "tentu", "tentulah", "tentunya", "tertentu",  
 "seterusnya", "tapi", "tetapi", "setiap", "tiap", "setidaknya",  
 "tidak", "tidakkah", "tidaklah", "toh", "waduh", "wah",  
 "wahai", "sewaktu", "walau", "walaupun",  
 "wong", "yaitu", "yakni", "yang", "untuk", "arah", "jalan",  
 "surabaya", "macet"))

### C.3 Stopword model 7 ketiga

"lancar", "terimakasih", "arah", "jalan", "light", "surabaya", "polisi", "infonya", "info", "macet", "hari", "tolong", "padat", "untuk"

### C.4 20 kata teratas yang akan dijadikan stopwords

positif		negatif		netral	
word	freq	word	freq	word	freq
lancar	44	jalan	128	jalan	371
ramai	22	macet	108	surabaya	218
terimakasih	18	arah	90	arah	211
good	16	padat	65	macet	190
arah	13	polisi	54	mohon	151
jalan	12	surabaya	52	terimakasih	120
terpantau	11	hari	42	mobil	117
lintas	10	jam	34	motor	116
light	8	pdam	34	hari	114
traffic	8	mati	30	polisi	113
semoga	7	raya	28	info	106
surabaya	7	air	27	tolong	104
kebakaran	6	lampu	27	padat	98
polisi	5	tolong	27	kawan	82
wonokromo	5	motor	26	lancar	78
alhamdulillah	4	jembatan	25	selamat	77
dolog	4	light	25	min	76
gubsuryo	4	unesa	24	daerah	74
infonya	4	pasar	23	pagi	70
kasih	4	petugas	23	mati	69



## LAMPIRAN D

### HASIL GRIDSEARCH SVM

#### D.1 Kernel Linear 3 Pembobotan

Berikut adalah hasil gridsearch untuk kernel Linear dengan 3 metode pembobotan dimana nilai gamma tidak akan memberikan pengaruh.

		GAM MA (Tf)			GAM MA (Tfidf)			GAM MA (bin)
COST		2 <sup>-1.50</sup>	COST		2 <sup>-1.50</sup>	COST		2 <sup>-1.50</sup>
	2 <sup>-20</sup>	78,777 39		2 <sup>-20</sup>	78,777 39		2 <sup>-20</sup>	78,777 39
	2 <sup>-19</sup>	78,777 39		2 <sup>-19</sup>	78,777 39		2 <sup>-19</sup>	78,777 39
	2 <sup>-18</sup>	78,777 39		2 <sup>-18</sup>	78,777 39		2 <sup>-18</sup>	78,777 39
	2 <sup>-17</sup>	78,777 39		2 <sup>-17</sup>	78,777 39		2 <sup>-17</sup>	78,777 39
	2 <sup>-16</sup>	78,777 39		2 <sup>-16</sup>	78,777 39		2 <sup>-16</sup>	78,777 39
	2 <sup>-15</sup>	78,777 39		2 <sup>-15</sup>	78,777 39		2 <sup>-15</sup>	78,777 39
	2 <sup>-14</sup>	78,777 39		2 <sup>-14</sup>	78,777 39		2 <sup>-14</sup>	78,777 39
	2 <sup>-13</sup>	78,777 39		2 <sup>-13</sup>	78,777 39		2 <sup>-13</sup>	78,777 39
	2 <sup>-12</sup>	78,777 39		2 <sup>-12</sup>	78,777 39		2 <sup>-12</sup>	78,777 39
	2 <sup>-11</sup>	78,777 39		2 <sup>-11</sup>	78,777 39		2 <sup>-11</sup>	78,777 39
	2 <sup>-10</sup>	78,777 39		2 <sup>-10</sup>	78,777 39		2 <sup>-10</sup>	78,777 39

	$2^{-9}$	78,777 39		$2^{-9}$	78,777 39		$2^{-9}$	78,777 39
	$2^{-8}$	78,777 39		$2^{-8}$	78,777 39		$2^{-8}$	78,777 39
	$2^{-7}$	78,777 39		$2^{-7}$	78,777 39		$2^{-7}$	78,777 39
	$2^{-6}$	78,777 39		$2^{-6}$	78,777 39		$2^{-6}$	78,777 39
	$2^{-5}$	78,777 39		$2^{-5}$	78,777 39		$2^{-5}$	78,777 39
	$2^{-4}$	78,66		$2^{-4}$	78,892 73		$2^{-4}$	78,777 39
	$2^{-3}$	77,16		$2^{-3}$	77,508 65		$2^{-3}$	76,816 61
	$2^{-2}$	75,54		$2^{-2}$	71,280 28		$2^{-2}$	74,394 46
	$2^{-1}$	73,47		$2^{-1}$	64,475 2		$2^{-1}$	72,779 7
	$2^{-0,9}$	73,24		$2^{-0,9}$	63,898 5		$2^{-0,9}$	72,433 68
	$2^{-0,8}$	73,01		$2^{-0,8}$	63,437 14		$2^{-0,8}$	72,895 04
	$2^{-0,7}$	72,77		$2^{-0,7}$	62,860 44		$2^{-0,7}$	72,895 04
	$2^{-0,6}$	72,77		$2^{-0,6}$	62,745 1		$2^{-0,6}$	73,010 38
	$2^{-0,5}$	72,66		$2^{-0,5}$	62,399 08		$2^{-0,5}$	73,125 72
	$2^{-0,4}$	73,01		$2^{-0,4}$	61,361 01		$2^{-0,4}$	73,241 06
	$2^{-0,3}$	72,54		$2^{-0,3}$	61,476 36		$2^{-0,3}$	72,779 7
	$2^{-0,2}$	72,43		$2^{-0,2}$	60,899 65		$2^{-0,2}$	72,664 36

2 <sup>-</sup> 0,1	72,08	2 <sup>-</sup> 0,1	61,014 99	2 <sup>-</sup> 0,1	72,318 34
2 <sup>0</sup>	71,97	2 <sup>0</sup>	60,668 97	2 <sup>0</sup>	72,318 34
2 <sup>0,1</sup>	71,97	2 <sup>0,1</sup>	60,553 63	2 <sup>0,1</sup>	72,203
2 <sup>0,2</sup>	71,85	2 <sup>0,2</sup>	60,553 63	2 <sup>0,2</sup>	71,972 32
2 <sup>0,3</sup>	71,85	2 <sup>0,3</sup>	60,438 29	2 <sup>0,3</sup>	72,203
2 <sup>0,4</sup>	71,51	2 <sup>0,4</sup>	60,207 61	2 <sup>0,4</sup>	72,087 66
2 <sup>0,5</sup>	71,28	2 <sup>0,5</sup>	60,322 95	2 <sup>0,5</sup>	72,087 66
2 <sup>0,6</sup>	71,28	2 <sup>0,6</sup>	60,207 61	2 <sup>0,6</sup>	71,972 32
2 <sup>0,7</sup>	71,04	2 <sup>0,7</sup>	60,322 95	2 <sup>0,7</sup>	71,626 3
2 <sup>0,8</sup>	71,04	2 <sup>0,8</sup>	60,322 95	2 <sup>0,8</sup>	71,049 6
2 <sup>0,9</sup>	70,47	2 <sup>0,9</sup>	59,746 25	2 <sup>0,9</sup>	71,049 6
2 <sup>1</sup>	70,58	2 <sup>1</sup>	59,054 21	2 <sup>1</sup>	71,164 94
2 <sup>2</sup>	68,97	2 <sup>2</sup>	58,823 53	2 <sup>2</sup>	69,319 49
2 <sup>3</sup>	69,43	2 <sup>3</sup>	59,054 21	2 <sup>3</sup>	68,512 11
2 <sup>4</sup>	69,08	2 <sup>4</sup>	57,900 81	2 <sup>4</sup>	68,627 45
2 <sup>5</sup>	69,08	2 <sup>5</sup>	58,246 83	2 <sup>5</sup>	68,627 45
2 <sup>6</sup>	69,08	2 <sup>6</sup>	58,246 83	2 <sup>6</sup>	68,627 45

	2 <sup>7</sup>	69,08		2 <sup>7</sup>	58,246 83		2 <sup>7</sup>	68,627 45
	2 <sup>8</sup>	69,08		2 <sup>8</sup>	58,246 83		2 <sup>8</sup>	68,627 45
	2 <sup>9</sup>	69,08		2 <sup>9</sup>	58,246 83		2 <sup>9</sup>	68,627 45
	2 <sup>10</sup>	69,08		2 <sup>10</sup>	58,246 83		2 <sup>10</sup>	68,627 45

## D.2 Kernel Linear Hanya Tf

GAMMA				
		2 <sup>-1.50</sup>	2 <sup>0</sup>	2 <sup>1.50</sup>
COST	2 <sup>-10</sup>	78,77	78,77	78,77
	2 <sup>-9</sup>	78,77	78,77	78,77
	2 <sup>-8</sup>	78,77	78,77	78,77
	2 <sup>-7</sup>	78,77	78,77	78,77
	2 <sup>-6</sup>	78,77	78,77	78,77
	2 <sup>-5</sup>	78,77	78,77	78,77
	2 <sup>-4</sup>	78,66	78,66	78,66
	2 <sup>-3</sup>	77,16	77,16	77,16
	2 <sup>-2</sup>	75,54	75,54	75,54
	2 <sup>-1</sup>	73,47	73,47	73,47
	2 <sup>-0,9</sup>	73,24	73,24	73,24
	2 <sup>-0,8</sup>	73,01	73,01	73,01
	2 <sup>-0,7</sup>	72,77	72,77	72,77
	2 <sup>-0,6</sup>	72,77	72,77	72,77
	2 <sup>-0,5</sup>	72,66	72,66	72,66
	2 <sup>-0,4</sup>	73,01	73,01	73,01
	2 <sup>-0,3</sup>	72,54	72,54	72,54

	$2^{-0,2}$	72,43	72,43	72,43
	$2^{-0,1}$	72,08	72,08	72,08
	$2^0$	71,97	71,97	71,97
	$2^{0,1}$	71,97	71,97	71,97
	$2^{0,2}$	71,85	71,85	71,85
	$2^{0,3}$	71,85	71,85	71,85
	$2^{0,4}$	71,51	71,51	71,51
	$2^{0,5}$	71,28	71,28	71,28
	$2^{0,6}$	71,28	71,28	71,28
	$2^{0,7}$	71,04	71,04	71,04
	$2^{0,8}$	71,04	71,04	71,04
	$2^{0,9}$	70,47	70,47	70,47
	$2^1$	70,58	70,58	70,58
	$2^2$	68,97	68,97	68,97
	$2^3$	69,43	69,43	69,43
	$2^4$	69,08	69,08	69,08
	$2^5$	69,08	69,08	69,08
	$2^6$	69,08	69,08	69,08
	$2^7$	69,08	69,08	69,08
	$2^8$	69,08	69,08	69,08
	$2^9$	69,08	69,08	69,08
	$2^{10}$	69,08	69,08	69,08

## D.3 Kernel RBF 3 Pembobotan

### D.3.1 Binary

		2^-0,4	2^-0,3	2^-0,2	2^-0,1	2^-0,09	2^-0,08	2^-0,07	2^-0,06	2^-0,05	2^-0,04	2^-0,03	2^-0,02	2^-0,01	2^0	2^0,01	2^0,02	2^0,03
gamma (bin)	2^-1.65	79,23875	79,70012	79,46943	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943
	2^-1.60	79,23875	79,70012	79,46943	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943
	2^-1.55	79,12341	79,46943	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943
	2^-1.50	79,12341	79,35409	79,70012	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943
	2^-1.45	79,12341	79,35409	79,70012	79,70012	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,35409
	2^-1.40	79,12341	79,12341	79,58478	79,70012	79,70012	79,70012	79,70012	79,70012	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,35409	79,35409	79,35409
	2^-1.35	79,00807	79,12341	79,46943	79,81546	79,81546	79,81546	79,81546	79,70012	79,70012	79,58478	79,58478	79,46943	79,35409	79,35409	79,35409	79,35409	79,46943
	2^-1.30	79,00807	79,12341	79,23875	79,81546	79,81546	79,81546	79,81546	79,81546	79,70012	79,58478	79,46943	79,70012	79,58478	79,58478	79,58478	79,58478	79,58478
	2^-1.25	79,00807	79,70012	79,23875	79,46943	79,46943	79,58478	79,58478	79,81546	79,70012	79,58478	79,58478	79,70012	79,70012	79,70012	79,70012	79,70012	79,58478
	2^-1.20	79,00807	79,70012	79,23875	79,35409	79,35409	79,35409	79,35409	79,35409	79,46943	79,35409	79,46943	79,58478	79,46943	79,70012	79,70012	79,70012	79,70012
	2^-1.15	79,00807	79,00807	79,00807	79,23875	79,23875	79,12341	79,12341	79,23875	79,35409	79,23875	79,23875	79,46943	79,23875	79,46943	79,46943	79,46943	79,58478
	2^-1.10	79,00807	79,00807	79,00807	79,00807	79,00807	79,12341	79,12341	79,23875	79,23875	79,12341	79,12341	79,12341	79,23875	79,35409	79,35409	79,35409	79,35409
	2^-1.05	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,12341	79,12341	79,12341	79,12341
	2^-1	79,00807	79,00807	79,00807	79,00807	79,00807	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807

	COST																
	2^0.04	2^0.05	2^0.06	2^0.07	2^0.08	2^0.09	2^0.1	2^0.2	2^0.3	2^0.4	2^0.5	2^0.6	2^0.7	2^0.8	2^0.9	2^1	2^2
2^1.65	79,46943	79,58478	79,70012	79,70012	79,70012	79,70012	79,58478	79,23875	79,58478	79,35409	79,12341	79,35409	79,46943	79,23875	79,23875	79,23875	79,23875
2^1.60	79,46943	79,46943	79,46943	79,58478	79,58478	79,46943	79,23875	79,23875	79,23875	79,12341	79,00807	79,00807	79,35409	79,23875	79,23875	79,23875	79,23875
2^1.55	79,46943	79,46943	79,46943	79,35409	79,35409	79,23875	79,23875	79,12341	79,23875	79,12341	79,00807	79,00807	79,23875	79,23875	79,23875	79,23875	79,23875
2^1.50	79,46943	79,35409	79,35409	79,35409	79,23875	79,23875	79,23875	79,23875	79,23875	79,12341	79,00807	79,00807	79,23875	79,12341	79,23875	79,23875	79,23875
2^1.45	79,35409	79,35409	79,46943	79,35409	79,23875	79,23875	79,23875	79,35409	79,00807	79,12341	79,00807	79,00807	79,23875	79,12341	79,12341	79,00807	79,00807
2^1.40	79,46943	79,46943	79,46943	79,46943	79,23875	79,23875	79,23875	79,35409	79,12341	78,89273	78,77739	79,00807	79,23875	79,00807	78,89273	78,89273	78,77739
2^1.35	79,46943	79,46943	79,46943	79,46943	79,35409	79,23875	79,23875	79,23875	79,12341	78,89273	78,77739	78,77739	78,89273	78,89273	78,89273	78,89273	78,77739
2^1.30	79,58478	79,58478	79,46943	79,46943	79,46943	79,35409	79,35409	79,23875	79,35409	78,89273	78,89273	78,77739	78,89273	78,89273	78,89273	78,89273	78,77739
2^1.25	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,23875	79,35409	79,12341	78,89273	78,77739	78,89273	78,89273	78,89273	78,89273	78,77739
2^1.20	79,70012	79,70012	79,70012	79,70012	79,58478	79,58478	79,58478	79,23875	79,23875	79,12341	78,89273	78,89273	78,89273	79,00807	79,00807	78,89273	78,89273
2^1.15	79,58478	79,70012	79,70012	79,70012	79,70012	79,70012	79,70012	79,35409	79,23875	79,35409	79,23875	79,23875	78,89273	78,89273	79,00807	79,00807	78,89273
2^1.10	79,46943	79,46943	79,46943	79,46943	79,58478	79,58478	79,58478	79,35409	79,35409	79,23875	79,12341	79,12341	79,00807	79,00807	79,00807	79,12341	78,89273
2^1.05	79,23875	79,35409	79,35409	79,35409	79,35409	79,35409	79,46943	79,35409	79,35409	79,35409	79,35409	79,12341	79,00807	79,00807	79,00807	79,12341	79,00807
2^1	79,00807	79,00807	79,12341	79,12341	79,12341	79,23875	79,35409	79,46943	79,35409	79,35409	79,35409	79,23875	79,23875	79,12341	79,00807	79,00807	79,12341
	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^10									
2^1.65	79,23875	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341									
2^1.60	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341									
2^1.55	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341									
2^1.50	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341									
2^1.45	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273									
2^1.40	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.35	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.30	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.25	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.20	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.15	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.10	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739									
2^1.05	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273									
2^1	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807									

[illegible]

	COST																		
	2 <sup>o</sup> 0.04	2 <sup>o</sup> 0.05	2 <sup>o</sup> 0.06	2 <sup>o</sup> 0.07	2 <sup>o</sup> 0.08	2 <sup>o</sup> 0.09	2 <sup>o</sup> 0.1	2 <sup>o</sup> 0.2	2 <sup>o</sup> 0.3	2 <sup>o</sup> 0.4	2 <sup>o</sup> 0.5	2 <sup>o</sup> 0.6	2 <sup>o</sup> 0.7	2 <sup>o</sup> 0.8	2 <sup>o</sup> 0.9	2 <sup>o</sup> 1	2 <sup>o</sup> 2	2 <sup>o</sup> 3	2 <sup>o</sup> 4
2 <sup>o</sup> -1.65	79,23875	79,23875	79,23875	79,23875	79,12341	79,12341	79,23875	79,12341	79,12341	79,00807	79,23875	79,12341	79,12341	79,12341	78,23875	79,35409	79,35409	79,35409	79,35409
2 <sup>o</sup> -1.60	79,12341	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,12341	79,12341	79,00807	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,23875	79,23875
2 <sup>o</sup> -1.55	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,23875	79,23875	79,12341	79,12341	79,00807	79,00807	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341
2 <sup>o</sup> -1.50	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,23875	79,23875	79,12341	79,12341	79,12341	79,00807	79,00807	79,12341	79,12341	79,12341	79,12341	79,00807	79,00807
2 <sup>o</sup> -1.45	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,00807	79,89273	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2 <sup>o</sup> -1.40	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,00807	79,89273	79,89273	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2 <sup>o</sup> -1.35	79,00807	79,00807	79,00807	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,00807	79,00807
2 <sup>o</sup> -1.30	79,89273	79,89273	79,89273	79,89273	79,00807	79,00807	79,00807	79,12341	79,00807	79,12341	79,12341	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273
2 <sup>o</sup> -1.25	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,00807	79,12341	79,00807	79,12341	79,12341	79,77739	79,89273	79,89273	79,00807	79,00807	79,00807	79,00807
2 <sup>o</sup> -1.20	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,00807	79,12341	79,00807	79,00807	79,12341	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273	79,89273
2 <sup>o</sup> -1.15	79,00807	79,00807	79,00807	79,00807	79,89273	79,89273	79,89273	79,89273	79,89273	79,00807	79,12341	79,00807	79,00807	79,89273	79,77739	79,77739	79,77739	79,77739	79,77739
2 <sup>o</sup> -1.10	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,89273	79,00807	79,00807	79,89273	79,00807	79,89273	79,77739	79,77739	79,77739	79,77739	79,77739	79,77739
2 <sup>o</sup> -1.05	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,89273	79,89273	79,00807	79,00807	79,89273	79,77739	79,86205	79,86205	79,86205	79,86205	79,86205	79,86205
2 <sup>o</sup> -1	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,89273	79,89273	79,00807	79,00807	79,00807	79,00807	79,77739	79,86205	79,86205	79,86205	79,86205	79,86205



	2^5	2^6	2^7	2^8	2^9	2^10
2^-1.65	79,35409	79,35409	79,35409	79,35409	79,35409	79,35409
2^-1.60	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875
2^-1.55	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341
2^-1.50	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.45	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.40	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.35	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.30	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273
2^-1.25	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.20	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273
2^-1.15	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739
2^-1.10	78,77739	78,77739	78,77739	78,77739	78,77739	78,77739
2^-1.05	78,66205	78,66205	78,66205	78,66205	78,66205	78,66205
2^-1	78,66205	78,66205	78,66205	78,66205	78,66205	78,66205

### D.3.3 Tf

gamma (tf)		2 <sup>~-0.4</sup>	2 <sup>~-0.3</sup>	2 <sup>~-0.2</sup>	2 <sup>~-0.1</sup>	2 <sup>~-0.09</sup>	2 <sup>~-0.08</sup>	2 <sup>~-0.07</sup>	2 <sup>~-0.06</sup>	2 <sup>~-0.05</sup>	2 <sup>~-0.04</sup>	2 <sup>~-0.03</sup>	2 <sup>~-0.02</sup>	2 <sup>~-0.01</sup>	2 <sup>0</sup>	2 <sup>0.01</sup>	2 <sup>0.02</sup>	2 <sup>0.03</sup>	
	2 <sup>~-1.65</sup>	79,12341	79,46943	79,35409	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	
	2 <sup>~-1.60</sup>	79,12341	79,35409	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	
	2 <sup>~-1.55</sup>	79,00807	79,35409	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	
	2 <sup>~-1.50</sup>	79,00807	79,23875	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,58478	79,58478	
	2 <sup>~-1.45</sup>	79,00807	79,12341	79,58478	79,58478	79,58478	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,58478	79,58478	79,58478	79,58478	
	2 <sup>~-1.40</sup>	79,00807	79,00807	79,23875	79,70012	79,70012	79,70012	79,58478	79,58478	79,58478	79,46943	79,46943	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	
	2 <sup>~-1.35</sup>	78,89273	79,00807	79,23875	79,70012	79,70012	79,70012	79,70012	79,70012	79,70012	79,81546	79,70012	79,70012	79,70012	79,70012	79,58478	79,58478	79,58478	
	2 <sup>~-1.30</sup>	78,89273	79,00807	79,12341	79,58478	79,70012	79,70012	79,70012	79,70012	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	79,70012	79,70012	79,70012	
	2 <sup>~-1.25</sup>	78,89273	78,89273	79,12341	79,23875	79,35409	79,35409	79,58478	79,70012	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	
	2 <sup>~-1.20</sup>	78,89273	78,89273	79,12341	79,12341	79,23875	79,23875	79,35409	79,35409	79,46943	79,58478	79,58478	79,58478	79,70012	79,81546	79,81546	79,81546	79,81546	
	2 <sup>~-1.15</sup>	78,89273	78,89273	78,89273	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,35409	79,58478	79,58478	79,58478	79,58478	
	2 <sup>~-1.10</sup>	78,89273	78,89273	78,89273	79,12341	79,12341	79,12341	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,23875	79,35409	79,35409	
	2 <sup>~-1.05</sup>	78,89273	78,89273	78,89273	79,12341	79,12341	79,12341	79,23875	79,23875	79,12341	79,12341	79,12341	79,12341	79,12341	79,23875	79,23875	79,23875	79,23875	
COST																			
	2 <sup>0.04</sup>	2 <sup>0.05</sup>	2 <sup>0.06</sup>	2 <sup>0.07</sup>	2 <sup>0.08</sup>	2 <sup>0.09</sup>	2 <sup>0.1</sup>	2 <sup>0.2</sup>	2 <sup>0.3</sup>	2 <sup>0.4</sup>	2 <sup>0.5</sup>	2 <sup>0.6</sup>	2 <sup>0.7</sup>	2 <sup>0.8</sup>	2 <sup>0.9</sup>	2 <sup>1</sup>	2 <sup>2</sup>	2 <sup>3</sup>	2 <sup>4</sup>
2 <sup>~-1.65</sup>	79,46943	79,46943	79,46943	79,46943	79,46943	79,46943	79,58478	79,46943	79,46943	79,23875	79,23875	79,35409	79,58478	79,35409	79,23875	79,23875	79,23875	79,12341	79,12341
2 <sup>~-1.60</sup>	79,46943	79,46943	79,46943	79,58478	79,58478	79,58478	79,58478	79,23875	79,46943	79,23875	79,23875	79,23875	79,58478	79,35409	79,23875	79,23875	79,23875	79,12341	79,12341
2 <sup>~-1.55</sup>	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,23875	79,35409	79,23875	79,23875	79,23875	79,35409	79,35409	79,23875	79,23875	79,00807	79,00807	79,00807
2 <sup>~-1.50</sup>	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,35409	79,23875	79,12341	79,12341	79,35409	79,12341	79,23875	79,12341	79,00807	79,00807	79,00807
2 <sup>~-1.45</sup>	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,46943	79,35409	79,00807	79,00807	79,23875	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807
2 <sup>~-1.40</sup>	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,35409	79,23875	79,00807	79,00807	79,23875	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807
2 <sup>~-1.35</sup>	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,58478	79,35409	79,46943	79,23875	79,00807	79,00807	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807
2 <sup>~-1.30</sup>	79,70012	79,70012	79,70012	79,58478	79,58478	79,58478	79,58478	79,35409	79,35409	79,23875	79,23875	79,00807	79,23875	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807
2 <sup>~-1.25</sup>	79,81546	79,81546	79,70012	79,70012	79,70012	79,70012	79,70012	79,35409	79,35409	79,23875	79,12341	79,00807	79,12341	79,12341	79,00807	78,89273	78,89273	78,89273	78,89273
2 <sup>~-1.20</sup>	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	79,81546	79,46943	79,35409	79,35409	79,35409	79,12341	79,12341	79,12341	79,23875	79,23875	79,00807	78,89273	78,89273
2 <sup>~-1.15</sup>	79,70012	79,70012	79,81546	79,81546	79,81546	79,81546	79,81546	79,58478	79,46943	79,35409	79,23875	79,35409	79,23875	79,12341	79,12341	79,23875	79,12341	79,00807	79,00807
2 <sup>~-1.10</sup>	79,46943	79,58478	79,58478	79,58478	79,58478	79,70012	79,70012	79,81546	79,46943	79,46943	79,23875	79,23875	79,12341	79,12341	79,12341	79,23875	79,12341	79,00807	79,00807
2 <sup>~-1.05</sup>	79,23875	79,35409	79,35409	79,35409	79,35409	79,46943	79,46943	79,70012	79,46943	79,46943	79,23875	79,23875	79,12341	79,12341	79,12341	79,23875	79,12341	79,00807	79,00807

	2^0,5	2^0,6	2^0,7	2^0,8	2^0,9	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^10
2^-1.65	79,23875	79,35409	79,58478	79,35409	79,23875	79,23875	79,23875	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341
2^-1.60	79,23875	79,23875	79,58478	79,35409	79,23875	79,23875	79,23875	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341	79,12341
2^-1.55	79,23875	79,23875	79,35409	79,35409	79,23875	79,23875	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.50	79,12341	79,12341	79,35409	79,12341	79,23875	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.45	79,00807	79,00807	79,23875	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.40	79,00807	79,00807	79,23875	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.35	79,00807	79,00807	79,12341	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.30	79,23875	79,00807	79,23875	79,12341	79,12341	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.25	79,23875	79,12341	79,00807	79,12341	79,12341	79,12341	79,00807	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273
2^-1.20	79,35409	79,12341	79,12341	79,12341	79,23875	79,23875	79,00807	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273	78,89273
2^-1.15	79,23875	79,35409	79,23875	79,12341	79,12341	79,23875	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.10	79,23875	79,23875	79,12341	79,12341	79,12341	79,23875	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807
2^-1.05	79,46943	79,23875	79,12341	79,12341	79,12341	79,23875	79,12341	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807	79,00807

#### D.4 Kernel RBF Hanya Tf

Keseluruhan tabel dengan metode Tf dengan  $C=2^{-15}$  sampai  $2^{10}$



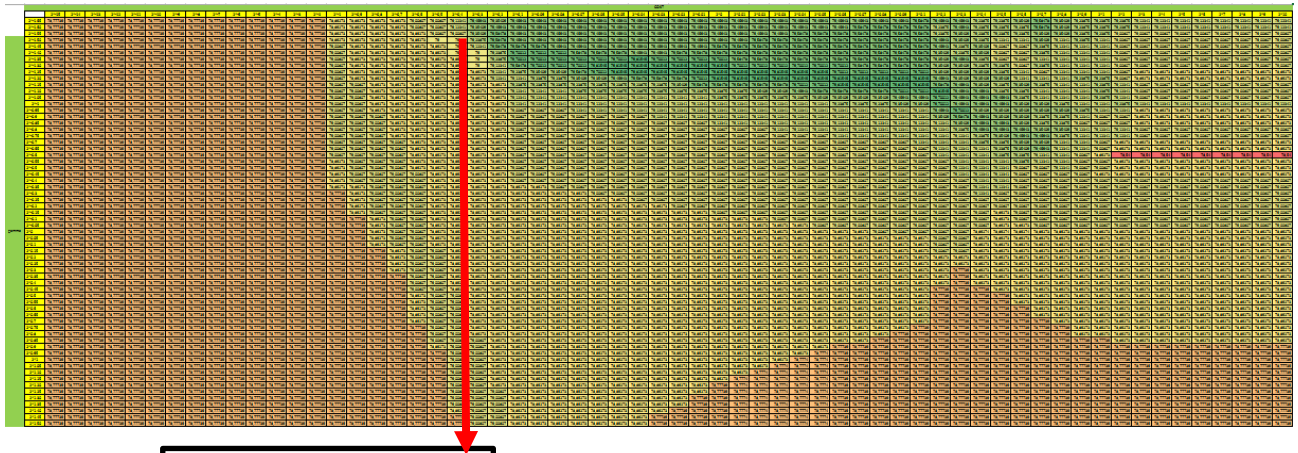
Nilai Akurasi yang didapat pada gambar yang diberi tanda panah adalah bernilai 78,77739







Nilai Akurasi yang  
ditandai pada gambar  
diatas adalah bernilai  
79,00807



Nilai Akurasi yang  
ditandai pada gambar  
diatas adalah bernilai 79

Hasil gridsearch yang berada di tengah, dengan  $C=2^{-0.7}$  sampai dengan  $C=2^{0.7}$  dan  $\gamma=2^{-1.65}$  sampai dengan  $2^{0.45}$

[illegible]







## LAMPIRAN E

### DAFTAR KORPUS YANG DIRUBAH

Berikut adalah daftar *corpus* yang dirubah pada data 1 dengan proses pembakuan setiap kalimatnya

1	+	rb	bernopol
2	2x	rmh	berpa
3	abu2	rs	besar2an
4	ad	ry	besuk
5	adl	sama2	bg
6	adlh	sampe	bgmn
7	aelamat	sampek	bgt
8	aj	satu2nya	bkn
9	aja	sbb	bl
10	akal2an	sbentar	blh
11	akhir2	sblm	blkg
12	akn	sby	blm
13	alhamd	sd	bln
14	alamat	sda	bnr
15	almt	sdg	bnyk
16	an	sdh	bnyu
17	anak2	sdiri	bp
18	ank	sdiskop	brati
19	ap	sdr	brcrmin
20	apa2	sdrhna	brg2
21	apbl	seh	brgkt
22	aplg	sellu	bribadah
23	aprtmn	sept	brisikan
24	as	sgera	brjaln

25	ato	sgl	brtanggungjwb
26	atoe	sgr	brulang
27	ats	sgra	bs
28	bantuany	sgt	bsa
29	baru2	shg	byk
30	bbrp	siap"	byr
31	bbtn	silarng	certifikat
32	bdn	sj	ciri"
33	berkali2	sjagar	ckp
34	ckup	stuju	dr
35	cm	sush	drh
36	cor2an	swhan	drpd
37	cp	sy	dskitr
38	cra	sya	dskt
39	d	t2	dtemukn
40	dah	tak	dtg
41	dahn	tau	dtng
42	ddt	tb2	du2k
43	dengn	td	dunk
44	dfr	tdk	emng
45	dg	telf	g
46	dger	telp	ga
47	dgn	temen2	gag
48	dikata2in	terakhr	gak
49	dilrang	terinjak2	gara"
50	dipasng	tgl	gara2
51	disblh	th	gg
52	dishub	thank	ggu
53	dk	thdp	gituh

54	dkung	thn	gk
55	dl	thnk	gmana
56	dll	thx	gmn
57	dln	tjd	gng
58	dlu	tk	gni
59	dmn	tl	gni
60	dmn2	tlah	gorong2
61	dng	tlg	grsk
62	dnyalakn	tlng	hati2
63	donk	tlp	hati2x
64	dpanggil2	tls	hj
65	dpet	tmn	hlang
66	dpn	tmn2	hln
67	kmrn	ucapn	lt
68	knp	ud	lwat
69	koq	uda	maceeeett
70	korupror	udah	main2
71	kpa	utk	makasih
72	kpd	wil	mantaff
73	krang	wkil	mbi
74	kr	wkt	mcet
75	krj	wrg	mctnya
76	krr	wrna	mdh
77	krna	y	menduga2
78	krt	yaa	menguadd
79	ksih	yah	menhub
80	kt	y	menrt
81	ktemu	prusuh	mer???sahkan
82	ktny	psn	met

83	ktr	psng	mewah2
84	kunang2	ptah	mhn
85	kyk	ptg	mhon
86	kyknya	ptgas	misl
87	lagi2	pjg	mjd
88	laki2	pke	mkan
89	lalin	pkl	mksh
90	lammmaaa	plg	mlm
91	lbh	pln"	mls
92	lg	pndaftarn	mmakai
93	lgi	pngaspalan	mmbrikan
94	lgsg	pnjang	mmg
95	lihtlh	pnjg	mmpunyai
96	ljr	pol	mnawarkan
97	lmyn	poldur	mnemukan
98	lnjt	pomp	mngatsinya
99	lsg	prnh	mninggal
100	sjk	hmbtan	jln
101	skali	hmpir	jm
102	skitar	hnda	jngan
103	skli	hndakny	k
104	skr	hny	k4
105	skrg	hnya	kac
106	sktr	hr	kalo
107	slamany	hrg	kanb
108	slamat	hrp	karna
109	slamt	hrpn	kata"
110	slkn	hrs	kawan2
111	sll	hrsnya	kawn

112	sllu	hrus	kbakaran
113	sm	ht2	kbr
114	sm2	hub	kbrnya
115	smoga	hubi	kcelakaan
116	smpe	hubla	kcil
117	smu	hubngi	kdg
118	smua	id	kebut2an
119	smua2nya	iii	kel
120	sp	ilmi	keluhan2
121	spd	infox	khan
122	spd mtr	internetx	khilnngn
123	spda	isiny	ki
124	speda mtr	istrahat	kita2
125	sprt	jaln	kjadian
126	sprti	jd	klo
127	spt	jdi	klr
128	spy	jg	klu
129	srg	jgn	kluarga
130	sring	jk	kmacetan
131	stiap	jl	kmarin
132	stlh	ylan	kmbang
133	tmpt	mnt	panas2
134	tnp	mntrari	pangsud
135	tnpa	moga	pasng
136	tny	motor2	pcr
137	tnya	mrasa	pd
138	tong2	mrh	pdahal
139	tp	mrk	pdhal
140	tpe	mrka	pdhl

141	tpi	ms	pd
142	trasa	msh	pemkot
143	trbaik	msih	pemsgn
144	trburu	mslh	per3an
145	trcela	msuk	per4an
146	trcinta	mudh	perjuanganny
147	trdpt	mw	permenhub
148	treimakasih	n	pershn
149	trhmbt	ndak	perum
150	trimakasih	neh	pg
151	trims	nfonya	pgalam
152	trjadi	ngaku2	pgwi
153	trkait	nggak	ptr
154	truk2	ni	pura2
155	trus	nmr	qt
156	trutama	no	r2
157	tsb	no plt	r4
158	ttd	nopol	radio2
159	ttg	nov	rakyat
160	ttp	nyalah	rambu2
161	tu	ok	
162	tulisanx	olh	
163	tw	org	
164	tx	org2	
165	u	p4an	



## DAFTAR LAMPIRAN

Berikut ini adalah lampiran dokumen dari penelitian ini. Dokumen-dokumen ini dapat dijadikan sebagai bukti dari pengerjaan penelitian ini.

KODE LAMPIRAN	LAMPIRAN
A	Kode Program
B	Daftar Kata Stopwords Bahasa Indonesia
C	<i>Wordcloud</i> masing-masing kelas
D	Hasil Gridsearch Metode SVM
E	Korpus yang dilakukan perubahan

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran terkait pengerjaan tugas akhir ini. Pada bagian kesimpulan, akan disimpulkan hasil pengerjaan tugas akhir ini. Sedangkan pada bagian saran, berisi saran yang berguna untuk pengembangan penelitian tugas akhir ini.

#### **7.1. Kesimpulan**

Setelah melakukan pengerjaan 2 skenario pada pengerjaan tugas akhir, maka dihasilkan beberapa kesimpulan yang dapat diambil yaitu :

1. Proses klasifikasi data twitter yang telah berhasil dibangun terbagi dalam 3 kelas sentimen yaitu positif, negatif, dan netral. Total data sampel tersebut dibagi menjadi dua jenis data, yakni 70% untuk data latih dan 30% untuk data uji pada setiap kelas klasifikasi.
2. Hasil penggunaan algoritma Naïve Bayes dengan menggunakan metode praproses penghapusan kata (*stopword removal*) mendapatkan hasil yang lebih tinggi dibanding yang tidak menggunakan. Untuk algoritma Support Vector Machine (SVM) metode praproses penghapusan kata (*stopword removal*) tidak terlalu berpengaruh terhadap hasil akurasi.
3. Dari 2 metode algoritma yang digunakan pada tahap klasifikasi yaitu Naïve Bayes dan Support Vector Machine (SVM) didapatkan hasil bahwa nilai akurasi yang lebih tinggi dikerjakan dengan metode Support Vector Machine (SVM). Untuk perbandingan penggunaan kernel yaitu

- linear dan RBF, kernel RBF mempunyai hasil yang lebih bagus. Untuk nilai  $C$  dan  $\gamma$  terbaik yaitu  $C=2^0$  dan  $\gamma=2^{-1,25}$ .
4. Model klasifikasi terbaik dengan menggunakan Support Vector Machine (SVM) yang dibangun dengan menggunakan data latih menghasilkan klasifikasi dengan nilai akurasi, *precision*, *recall*, dan *f-measure* berturut-turut pada kelas netral sebesar 78,66%, 99,85%, 79,67%, dan 88,6%. Nilai pada kelas netral memiliki nilai yang paling baik diantara positif dan negatif dikarenakan memang kelas netral mempunyai banyak anggota.
  5. Pada data yang memuat hanya tentang kinerja pelayanan publik Kota Surabaya cara pengerjaan klasifikasi terbaik yaitu dengan menggunakan Support Vector Machine (SVM) pada kernel Linear dan tanpa dilakukan stopword pada tahapan praprosesnya. Untuk nilai akurasinya yaitu sebesar 67,05% dan nilai *f-measure* untuk setiap kelas positif, negatif dan netral berturut-turut yaitu 50%, 70,96%, dan 63,99%.
  6. Visualisasi yang ditampilkan dalam penelitian ini berupa hasil wordcloud dan grafik streamgraph. Untuk wordcloud yang dibuat yaitu tentang data keseluruhan(2888 data) dan data yang hanya tentang Surabaya(293 data). Wordcloud yang dibuat pada data keseluruhan akan berpengaruh dalam pembuatan model klasifikasi Naïve Bayes.

## 7.2. Saran

Dalam pengerjaan tugas akhir ini masih terdapat kekurangan yang perlu untuk diperbaiki. Saran ini dapat digunakan untuk melakukan penelitian selanjutnya agar hasil yang di dapat lebih bagus. Berikut

adalah beberapa saran yang bisa digunakan dalam melakukan pengembangan penelitian ke depan yang lebih baik.

1. Proporsi data yang digunakan diusahakan seimbang antar masing-masing kelasnya agar seimbang dan dapat mempengaruhi peningkatan nilai akurasi.
2. Melakukan tahapan praproses yang lainnya seperti stemming dengan penggunaan algoritma Naïve Bayes.
3. Percobaan dalam mengganti pembagian proporsi penggunaan data latih dan data uji.

## DAFTAR PUSTAKA

- [1] K. P. A. N. d. R. B. R. Indonesia, "Pedoman Pemanfaatan Media Sosial Instansi Pemerintah," p. 7, 2012.
- [2] P. and A. (. P. J. I. I. , "Profil Pengguna Internet Indonesia 2014," 2014.
- [3] H. r. Gilman, G. Bullen, L. McCann and A. Black, "Discover an open source world," opensource.com, 20 Mei 2014. [Online]. Available: <http://opensource.com/government/14/5/3-ways-public-sector-wins-over-private-opengov>. [Accessed 23 September 2015].
- [4] U. Choirunnisa, "Mebiso.com," 3 November 2014. [Online]. Available: <http://mebiso.com/6-cara-meningkatkan-feedback-pelanggan-bagi-perusahaan/>. [Accessed 23 September 2015].
- [5] "Sosmedtoday," [Online]. Available: <http://www.sosmedtoday.com>. [Accessed 29 September 2015].
- [6] H. K. Surabaya, "Humas Surabaya," 12 Januari 2015. [Online]. Available: <http://humas.surabaya.go.id/index.php?option=news&det=548>. [Accessed 30 September 2015].
- [7] L. D Monte, "Business Standard," [Online]. Available: <http://www.business-standard.com/article/technology>. [Accessed 28 September 2015].
- [8] Skdi, "Dinas Komuniaksi dan Informatika Kota Surabaya," 24 September 2012. [Online]. Available: [http://dinkominfo.surabaya.go.id/dki.php?hal=detail\\_berita&id\\_berita=123](http://dinkominfo.surabaya.go.id/dki.php?hal=detail_berita&id_berita=123). [Accessed 2 Oktober 2015].

- [9] "KampungMedia," [Online]. Available: <http://www.kampungmedia.com/cgi-sys/suspendedpage.cgi>. [Accessed 2 Oktober 2015].
- [1 A. N. Farahiyah, "Perilaku Pemanfaatan Akun Twitter((Studi  
0] Deskriptif Tentang Perilaku Pemanfaatan Akun Twitter @e100ss Sebagai Media Pemenuhan Kebutuhan Informasi Pengguna Jalan Raya di Kota Surabaya)," vol. 3, no. 2, p. 3, 2014.
- [1 KBBI. [Online]. Available: <http://kbbi.web.id/klasifikasi>.  
1] [Accessed 28 September 2015].
- [1 [Online]. Available:  
2] <http://library.binus.ac.id/eColls/eThesisd/doc/Bab2DOC/2013-1-00500-SI%20Bab2001.doc..> [Accessed 25 September 2015].
- [1 R. J. Mooney, "Machine Learning text Categorization," p. 2,  
3] 2006.
- [1 A. J. Arriawati, I. Santoso and Y. Christyono. [Online].  
4] Available: <http://core.ac.uk/download/pdf/11724347.pdf>. [Accessed 23 September 2015].
- [1 Y. Zhao, R and Data Mining: Examples and Case Studies 1,  
5] Elsevier, 2012.
- [1 P. N. Tan, M. Steinbach and V. Kumar, Introduction to Data  
6] Mining, Pearson.
- [1 J. Han, M. Kamber and J. Pei, Data Mining Concepts and  
7] Techniques, 2011.
- [1 H. Kridalaksana, Kamus Linguistik, Gramedia Pustaka  
8] Utama, 2008.
- [1 M. Y. Nur and D. D. Santika, "ANALISIS SENTIMEN  
9] PADA DOKUMEN BERBAHASA INDONESIA DENGAN PENDEKATAN SUPPORT VECTOR MACHINE," *Konferensi Nasional Sistem dan Informatika*, p. 2, 2011.

- [2 M. H. Ch, "Text Mining," [Online]. Available:  
0] <http://iwanarif.lecturer.pens.ac.id/kuliah/dm/6Text%20Mining.pdf>. [Accessed 23 September 2015].
- [2 C. K, "Bayesian Online Classifiers for Text Classification and  
1] Filterin," [Online]. Available:  
<http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf>. [Accessed 25 September 2015].
- [2 A. Tamilselvi and M. P. Taj, "Sentiment Analysis of Micro  
2] blogs using Opinion Mining Classification Algorithm," *International Journal of Science and Research (IJSR)*, vol. 2, no. 10, p. 4, 2013.
- [2 N. M. Ari Lestari, I. K. G. Darma Putra and A. K. Agung  
3] Cahyawan, "Personality Types Classification for Indonesian Text in Partners Searching Website Using Naïve Bayes Methods," *International Journal of Computer Science Issues*, vol. 10, no. 1, 2013.
- [2 A. S. Nugroho, A. B. Witarto and D. Handoko, " Support  
4] Vector Machine - Teori dan Aplikasi dalam Bioinformatika," 2003. [Online]. Available:  
<http://asnugroho.net/papers/ikcsvm.pdf>. [Accessed 6 Oktober 2015].
- [2 M. N. Pamungkas, "Academia Edu," IBI Darmajaya,  
5] [Online]. Available:  
[https://www.academia.edu/6435256/Support\\_Vector\\_Machine](https://www.academia.edu/6435256/Support_Vector_Machine). [Accessed 12 Desember 2015].
- [2 E. E. Pratama and B. R. Trilaksono, "Klasifikasi Keluhan  
6] Pelanggan Berdasarkan Tweet dengan Menggunakan Metode Support Vector Machine (SVM)," Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung, 2014.
- [2 C. Hsu, C. Chang and C. Lin, "A Practical Guide to Support  
7] Vector Classification," Departement of Computer Science National Taiwan University, Taipe, 2010.

- [2 P. Flach, Machine Learning: The Art and Science of  
8] Algorithms that Make Sense of Data, 2012.
- [2 L. Binus, "Tesis," p. 48.  
9]
- [3 D. F. Priyambada, Rancang Bangun Aplikasi Backend dan  
0] Web Services Penyampaian Laporan Masyarakat Berbasis  
Crowdsourcing Melalui Jejaring Sosial Twitter, Surabaya:  
ITS Surabaya, 2016.
- [3 Nurandi, "Github," [Online]. Available:  
1] <https://raw.githubusercontent.com/nurandi/nurandi.net/master/data/stopwords-id.txt>. [Accessed 10 Maret 2016].
- [3 I. Fellows, "Package 'wordcloud'," 20 Februari 2015.  
2] [Online]. Available: <https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf>.  
[Accessed 10 April 2016].
- [3 E. Neuwirth, "Package 'RColorBrewer'," 19 Februari 2015.  
3] [Online]. Available: <https://cran.r-project.org/web/packages/RColorBrewer/RColorBrewer.pdf>.  
[Accessed 10 April 2016].
- [3 hadley, "Package 'dplyr'," 1 September 2015. [Online].  
4] Available: <https://cran.r-project.org/web/packages/dplyr/dplyr.pdf>. [Accessed 15  
April 2016].
- [3 hrbrmstr, "Introduction to the streamgraph htmlwidgtet R  
5] Package," 24 Desember 2014. [Online]. Available:  
<https://github.com/hrbrmstr/streamgraph>.



## RIWAYAT PENULIS



Nuke Yulnida Aden Faradhillah dilahirkan di Surabaya pada tanggal 1 Juli 1994. Gadis yang disapa Fara ini menghabiskan waktu kecilnya di Gresik, menempuh pendidikan dari TK hingga SMP di Surabaya, yakni SD Laboratorium UNESA, dan SMPN 12 Surabaya. Jenjang pendidikan berikutnya Fara memilih untuk

menempuh di SMA Negeri 18 Surabaya dan lulus pada tahun 2012. Pada tahun yang sama, Fara melanjutkan pendidikan ke bangku perguruan tinggi di Jurusan Sistem Informasi FTIf Institut Teknologi Sepuluh Nopember Surabaya. Di jurusan tersebut penulis terdaftar dengan NRP 5212100123.

Aktivitasnya selain kuliah cukup beragam dengan mengikuti keanggotaan kepanitiaan kegiatan di ITS. Pada tahun pertamanya Fara sempat mendapatkan kesempatan untuk megang di HMSI sebagai sekertaris. Sedangkan pada tahun keduanya Fara menjadi staff di organisasi HMSI ITS sebagai staff Biro Kesekretariatan. Kesempatan lain datang pada tahun ketiga, Fara memutuskan untuk menjadi bendahara 2 di organisasi HMSI ITS. Selain itu, selama menyandang gelar sebagai mahasiswa, Fara menyempatkan diri menjadi asisten praktikum beberapa matakuliah. Untuk bidang minat tugas akhir yang dikerjakan, Fara memilih Akuisisi Data dan Diseminasi Informasi sebagai bidang minatnya karena ketertarikannya dalam bidang pengolahan data dan informasi serta visualisasi. Penulis dapat dihubungi melalui email [nukeyulnidaaf@gmail.com](mailto:nukeyulnidaaf@gmail.com) untuk keperluan penelitian.